

**НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
ДЕПАРТАМЕНТ МУЗИКА**

ас. Стела Атанасова

ДИСЕРТАЦИЯ

**ИЗСЛЕДВАНЕ НА МЕТОДИ И
СРЕДСТВА ЗА СЪЗДАВАНЕ НА
КОМПЮТЪРНА АЛГОРИТМИЧНА
МУЗИКА**

**представена за присъждане на
образователна и научна степен “Доктор”**

**Научен ръководител: проф. д-р Симо
Лазаров**

София, 2011

Съдържание

УВОД	5
ГЛАВА 1. ИСТОРИЧЕСКО ИЗСЛЕДВАНЕ ВЪРХУ РАЗВИТИЕТО НА КОМПЮТЪРНАТА И АЛГОРИТМИЧНА МУЗИКА	10
1.1 Компютърна музика	12
1.1.1 Компютърна музика - определение	14
1.1.2 Исторически обзор върху развитието на компютърната музика.....	17
1.1.3 Музика и нови технологии	18
1.1.4 СИНТЕЗ НА ЗВУК ЗА КОМПЮТЪРНА КОМПОЗИЦИЯ	21
1.2 Алгоритмична музика	22
1.2.1 Алгоритмична музика - определение	23
1.2.2 Историческо изследване върху развитието на алгоритмичната музика .	24
1.2.3 КОМПОЗИТОРИ ИЗПОЛЗВАЩИ ПРИНЦИПА НА НЕСЪЗНАТЕЛНО АЛГОРИТМИЧНО КОМПОЗИРАНЕ	38
1.2.4 СЪВРЕМЕННИ ТЕХНИКИ.....	44
1.3 Фрактална музика.....	48
1.3.1 Същност на фракталната музика	50
1.3.2 Изчертаване на музика (кодово-графично преобразуване).....	51
1.3.3 двоичният код на нотното изразяване	51
1.3.4 ОТ MIDI КЪМ 12 ТЕТ (<i>TONE EQUAL TEMPERAMENT</i>).....	52
ПРИНОСИ.....	55
ГЛАВА 2. НОВИ ПОДХОДИ ЗА СЪЗДАВАНЕ НА КОМПЮТЪРНА МУЗИКА	56
2.1 АНАЛИЗ НА НОВИТЕ ПОХВАТИ ЗА СЪЗДАВАНЕ НА КОМПЮТЪРНА МУЗИКА	58
2.1.1 ДИГИТАЛЕН СИНТЕЗ НА МУЗИКАТА.....	58
2.1.2 МУЗИКАТА НА КОМПЮТЪРНИТЕ ТЕХНОЛОГИИ.....	60
2.1.3 Концепция за създаването на дигитална музика чрез компютърните технологии	61
2.1.4 Нови методи и подходи при създаването на компютърна музика	62
2.1.5 Параметричен контрол при синтез на музиката.....	64
2.2 Кодово-графично преобразуване на музика при представяне на многомерни данни.....	67
2.2.1 Влияние на различните характеристики при процеса на възприятие	68

2.2.2 Възможности за представяне на многомерни данни със звук	70
2.2.3 Кодово-графично преобразуване на звуци към събития	72
2.2.4 Обобщение на световните практики.....	73
ПРИНОСИ.....	75
ГЛАВА 3. ФРАКТАЛНОСТТА КАТО МОДЕЛ ПРИ КОМПОЗИРАНЕТО НА МУЗИКА	76
3.1 МОДЕЛНОСТ НА ФРАКТАЛНАТА МУЗИКА	76
3.2 Кратка история на фракталите	78
3.3 ПОВЕДЕНЧЕСКИ МОДЕЛ И СХОДНОСТ МЕЖДУ МУЗИКАТА И ФРАКТАЛИТЕ	80
3.3.1 СТАНДАРТНИ ФОРМИ	81
3.4 ПОВТОРЕНИЕТО В МУЗИКАТА КАТО ФОРМА НА ФРАКТАЛЕН МОДЕЛ.....	83
3.4.1 ПРОЦЕСНА МУЗИКА.....	83
3.4.2 Повтаряемост в съвременната популярна музика.....	84
3.5 СЪВРЕМЕННИ МУЗИКАЛНИ ФОРМИ И ТЯХНОТО СХОДСТВО С ФРАКТАЛНИЯ МОДЕЛ	90
3.5.1 Свойството на фракталност в музиката	92
ПРИНОСИ.....	93
ГЛАВА 4. АНАЛИЗ НА МЕТОДИТЕ ЗА СЪЗДАВАНЕ НА АЛГОРИТМИЧНА МУЗИКА	94
4.1 Моделиране на традиционни, не-алгоритмични композиционни процедури	95
4.1.1 Моделиране на нови процедури.....	97
4.2 Избор на алгоритми от музикални дисциплини – методи за композиране на алгоритмична музика	97
4.2.1 L-системите (L-Systems)	97
4.2.2 КЛЕТЪЧНА АВТОМАТИЗАЦИЯ (Cellular automata)	108
4.2.3 ФРАКТАЛИ	117
4.2.4 МЕТОДИ ЗА СЪЗДАВАНЕ НА ФРАКТАЛНА МУЗИКА.....	122
4.2.5 МУЗИКАТА НА ФРАКТАЛИТЕ	124
4.2.6 ПРИЛОЖЕНИЕ НА ФРАКТАЛНОСТТА В МУЗИКАТА	126
4.2.7 Вериги на Марков.....	130

4.3 ТЕХНИКИ И ПРИЛОЖЕНИЕ НА МАТЕМАТИЧЕСКИ МОДЕЛИ В МУЗИКАТА.....	138
4.3.1 кодово-графични преобразуващи ТЕХНИКИ	138
4.3.2 Стохастични алгоритми (<i>STOCHASTIC ALGORITHMS</i>).....	141
4.3.3 МУЗИКА ОТ ХАОСА.....	146
4.3.4 АЛГОРИТМИ ЗА ПОЛУЧАВАНЕ НА ШУМ	165
4.3.5 ГЕНЕТИЧНИ АЛГОРИТМИ.....	168
4.3.6 ТЕХНИКИ ЗА КОМПОЗИРАНЕ НА АЛГОРИТМИЧНА МУЗИКА	174
4.3.7 Системата <i>variations</i>	194
4.3.8 ЦЕЛТА НА АЛГОРИТМИЧНОТО КОМПОЗИРАНЕ	199
ПРИНОСИ.....	202
ГЛАВА 5. СРЕДСТВА ЗА СЪЗДАВАНЕ НА КОМПЮТЪРНА АЛГОРИТМИЧНА МУЗИКА	204
ЗАКЛЮЧЕНИЕ	239
ОСНОВНИ ПРИНОСИ В ДИСЕРТАЦИЯТА	245
БИБЛИОГРАФИЯ	247
ПРИЛОЖЕНИЕ 1	259
ПРИЛОЖЕНИЕ 2	279
РЕЧНИК	298

УВОД

Музиката е свят, чиито граници преминават неизмеримата безкрайност на Вселената. Тя живее в пълна хармония със света на относителните неща и се създава постоянно. Музиката еволюира и се самогенерира. Тя е началото на края и края на началото. Безкрайността е в същността на нейния генезис и е в основата на формулата за нейното себеподобие.

Целта на настоящата дисертация е насочена изцяло към изследване генезиса на музиката и формулата за нейната еволюция, както и изследване на основните методи и средства, чрез които се създава компютърната алгоритмична музика¹. За постигането на тази цел са поставени за изпълнение няколко конкретни задачи:

1. Анализ на връзката между музиката и математиката.
2. Историческо изследване на развитието на компютърната и алгоритмичната музика.
3. Изследване на музикалния фрактал и фракталния модел в музиката.
4. Изследване и анализ върху основните методи, използвани за създаване на компютърна алгоритмична музика.
5. Изследване и анализ на основните средства, чрез които се създава компютърна алгоритмична музика.

Тезата, към която ще насоча вниманието ви в своята дисертация е, че алгоритмичната музика се създава на основата на един постоянен, рекурсивен процес на самогенериране и самонаподобяване. Това е процес, който наподобява процеса на генериране на фракталите. Фракталната музика е една необятна територия и в момента ние само вървим по повърхността ѝ. Фракталните композиции показват, че рекурсивните алгоритми могат да създават музика по същия начин, по който се създават фрактални изображения. Разликата е в това, че изходните данни не се преобразуват към пиксели, а се преобразуват към музикални стойности.

Никога музиката не е била толкова фрактална и разбрана като такава. Средствата и методите за създаване на алгоритмична и фрактална музика са стотици, но моето внимание бе привлечено не от самите похвати и способности за създаване на алгоритмична музика, а от факта, че чрез математиката – аз, класическият музикант - започнах да

¹ С термина алгоритмична музика се означава такава музика, в самото композиране на която съществено участие взимат автоматични устройства и компютърни програми /4.с.36/

разбирам музиката по съвсем друг начин. Бях провокирана да започна да изследвам самия генезис на музиката и да призная, че чрез методите и средствата на математиката, музиката се разкрива и отваря вратите към едно необятно и все по-будещо интерес научно пространство. Пространство, където музиката се създава с помощта на математиката и се самонаподобява, основавайки се на алгоритмични процеси.

За музиката няма време, нито предели, защото тя е част от Вселената и не може да бъде определена. Музиката е абстрактно изкуство, което излиза от границите на самото разбиране за музите и създава собствен свят, в който живее и създава настояще между очертанията на порядъка и абсолютния хаос. Безупречното разчертаване на света и подредбата на нещата в него отдавна вече не може да бъде обхванато с една теорема или един извод, от който да следва всичко останало. Музиката е началото на един безкраен свят, в който златното сечение може само да определи измеренията на безкрайните стойности, сред които живее музиката. Музиката е необятна и необхватна, неопределима, неизмерима и клони към безкрайността на положителните стойности . Тя е!

Живеем в бъдещето. То наложи нови истини, нови граници, нови правила . Бъдещето направи нови хора - с нови разбирания и светоусещания. Бъдещето създаде нова цивилизация, която разбира света и го отразява по съвсем различен начин. Ако в миналото изкуствата са били разбирани като мимезис - т.е. подражание на реалността и са създавали ценностни системи, отговарящи на материалния свят, в който са живяли обществата, то бъдещето сега, създаде свят, който да подражава самите изкуства.

Цикълът на еволюцията, който води света напред, е променлива стойност. Светът оцелява именно на базата на постоянната промяна, чрез която се самоусъвършенства. Светът е една константна промяна. Върви напред и все по-устремно нагоре. Води след себе си милиарди микросветове. Създава реалност, в която всеки един от тези негови микроподобия да съществува на точното място, в подходящото време. Светът оцелява именно, защото е изпълнен от самия себе си и подражава на всяко едно от своите подобия. А всяка една частица от него - създава свой собствен свят, в който живее, твори и се усъвършенства. Всяка една пращинка от този свят е началото на нов свят и продължението на този, който го е създал.

Това, което ни прави истински и непреходни, сред целия неистов устрем за промени и технологични разработки, е света на музиката. Тя е била, е и ще бъде част от този свят, както и той от нея. Нищо и никой не може да спре музиката - може

единствено да я промени. Никога повече няма да се родят гении като Моцарт, нито ще се създаде токата и фуга с мащабната мисъл на Бах. Всяко поколение идва с времето си и с търсенията си. Всяко изкуство се определя от търсенията на творците, които го създават и на публиката, която го преживява. Днешната масова публика няма да забележи раждането на гений от ранга на Бетховен, но ще създаде хиляди последователи, ако компютърен гений създаде музикален откъс, който да грабне останалите с електронните си семпли и ритъм. Много малко композитори си позволяват лукса да не са в ритъм с търсенията на хората, защото са творци и тяхна главна цел е да кажат нещо чрез музиката си и тя да бъде чута. Ако съвременният композитор не може да грабне аудиторията с класическите похвати, ще представи себе си чрез средствата на компютърните технологии. Защото днес масовата публика слуша и приема информацията от света по нов начин - чрез Интернет, чрез средствата за масова комуникация, чрез мобилни телефони и портативни компютри, чрез свръхмодерни компютърни и звукоозвучителни средства. Изискванията за това как да звучи един глас, за да достигне до масовия слушател и да грабне вниманието му вече са завишени до такава степен, че едно секундно смущение в ефира на дадена медия, ще накара масовия слушател или зрител да смени честотата или канала. Светът на глобализацията и модерните технологии - този, в който живеем, стана безпощаден към качеството на информацията, която подава към самите нас. Ние самите изискваме от себе си все повече - подтиквани от устрема си да вървим все по-напред и по-нагоре. Някога, някъде, някой - ще спре този процес. Но дотогава всички, които искат да са част от тази Вселена - ще вървят с темпото и в посоката, които налага прогреса .

Всичко днес се подчинява на закона – информацията да се предава със скоростта на светлината, във възможно най-редуциран и разбираем вид, да отговаря на сетивните възприятия, които имат нуждаещите се от нея, да се задържа възможно най-кратко в паметта им и да има способността да се самоизтрива. Но най-важното от всичко е – тази информация да успее да задържи вниманието на хората поне за няколко минути. Подчинявайки се на този закон, който налага жестоки правила, идва въпроса как едно музикално произведение ще успее да привлече и след това да задържи вниманието на хората към себе си?!

Логичното решение на този проблем е музиката да стане част от това главоземащо движение и в самата себе си да отрази хаоса и безпорядъка на света, в

който вече живеем. Така е започнала да се заражда и развива електроакустичната² музика, последвана от компютърната³ и електронна музика⁴, и вече налагаща една нова епоха в музиката – създаването на новата музика на 21 век – компютърната алгоритмична музика.

Началото на тази технологична епоха даде тласък на музиката да се себеизрази чрез начини и похвати, неизползвани досега, но даващи все по-неограничени възможности за откриване на нови музикални търсения. 21 век вече има нова класическа музика, която откри, че е част от самата себе си – и че всяка нейна малка частица наподобява цялото, както и цялото дава модел за подражание на милиардите нейни подобия. Музиката откри, че е себеподобна и в основата си съществува на базата на безкрайно повтарящи се и самогенериращи се елементи – наподобяващи структурата на фракталите. Всяка нова вариация е миниподобие на цялото, с което се предизвиква верига от все по-различни негови подобия. Музиката никога не е била по-фрактална и разбрана чрез принципите на фрактализация, отколкото се разбира, изследва и създава днес.

Затова днес музиката е нова, причудлива, правена по различен начин и звучаща все по-различно. Истината е, че светът днес чува музиката различно. И това я промени.

Един от факторите, които правят музиката толкова уникална и невъобразима, е нейната все така необяснима, но неоспорима връзка с математиката. Връзката между математиката и музиката вече не подлежи на обсъждане! Въпросът е доколко музиката и математиката си влияят и си помагат.

Математиката разшири възможностите на музиката през последните десетилетия, въпреки, че алгоритмичното композиране се използва от векове. Поради факта, че в днешно време вече се търси ново звучене, търсят се нови усещания и възприятия – това наложи самата музика да се промени и да започне да се преоткрива и да разширява възможностите си, използвайки математиката.

² Понятието електро-акустичен процес (процес използващ електричеството като принцип на звукоизвличане), ознаменува появата на нов вид енергия за звукоизвличане, приложим в музиката. Електронни музикални инструменти синтезират нов електро-акустичен аналог на традиционния акустичен звук. /3, с.8/

³ С термина “компютърна музика” се окачествява не само музиката, композирана от компютър, а всякакъв жанр музика, в чиято технология участва по някакъв начин и компютър. /6, с.23/

⁴ Електронната музика е събирателно понятие, което включва както звуковата форма, така и технологията за създаване на звука, така че независимо от дейността и функциите на различните центрове, тя няма строго определени граници /5, с.18/

Тази музика сега звучи странно, като всяко ново нещо. Но съм уверена, че след няколко години алгоритмичната музика ще застане редом до най-популярните стилове в музиката.

Алгоритъмът е редица от правила – той е като робот – съществува, докато изпълни зададените му правила. Той прави най-доброто, на което е способен. Но когато говорим за връзката му с музиката – той не може да бъде винен за музиката, която се създава вследствие на инструкциите, които са му подадени. Той не е музикант – не усеща и не чува музиката. Вярно е, че благодарение на програмите за алгоритмично композиране - всеки може да започне да композира музика, но не би могъл да разбере какво точно създава, ако не е наясно с основните принципи за композиране . Алгоритъмът не би могъл да създаде музика без основните характеристики, които я обуславят като такава.

Музиката и математиката имат еднакъв начин и подход на мислене – имат форма, която се запълва със съдържание. Всеки композитор всъщност решава една математическа задача, докато композира – дори той решава тази задача, докато чува музиката в себе си и след това я нотира.

Композиторът създава музика изчислявайки несъзнателно!

Защото чува музиката, несъзнателно, в съотношения. Чува музиката в интервали, в тризвучия и четиризвучия, в различни музикални форми и нотни трайности. А зад всичко това стоят числови съотношения!

Безупречното определение за математическия и музикален фрактал е библейският израз: “Аз съм Алфата и Омегата, началото и краят, първият и последният”.

ГЛАВА 1. ИСТОРИЧЕСКО ИЗСЛЕДВАНЕ ВЪРХУ РАЗВИТИЕТО НА КОМПЮТЪРНАТА И АЛГОРИТМИЧНА МУЗИКА

През цялата история на музиката, много композитори използват математическите модели като източник на композиционно творчество. Напоследък благодарение на компютрите композиторите могат да обединят сложните математически модели в композициите си без да е нужно да правят досадните и монотонни изчисления, които те изискват.

Една от целите на тази глава от дисертацията е да покаже, че математическите модели осигуряват за музикалните структури многовариантност и усъвършенстване. Като доказателство съм изложила в Приложение 1 важни факти свързани с връзката между музиката и математиката.

Композирането на музика с компютър постепенно зае своето заслужено място, наред с далеч по-традиционните начини за композиране, като създаването на творби върху нотния лист или импровизацията. Компютърната музика успя да навлезе и да приложи своите похвати в много музикални жанрове, включително поп, рок, техно, диско, джаз и филмова музика. Компютърната музика вече не е ограничена от естетиката, която се появява през 60-те и 70-те години и се основава изцяло на експерименталната работа с електронна⁵ и електроакустична музика. Времената, когато компютърно създадената музика можеше да се изпълнява в малко институции, използващи много скъпо оборудване, вече са минало. Появиха се нови категории елементи тангиращи с музикалните понятия: звуковите карти са вече стандартен елемент в компютрите, при това с все по-нарастващо качество; потребителският софтуер за музика улеснява композирането на музика чрез лесен за употреба графичен потребителски интерфейс, който в повечето случаи не изисква никакви способности за програмиране.

За да се разграничи специфичното в тази музика, за разлика от използването на електронни средства в други области и жанрове на музиката (филмова музика, театър, балет, попмузика, танцова музика и много други), нужно е да се подчертаят няколко неща които, които влизат в определението на съвременната електроакустична музика:

⁵ Самата електронна музика не е нов жанр, а принципно нова музика, която се различава от традиционната преди всичко по начина на звукоизвличане. /5, с.7/

- тя е артистична дейност, в която преобладава творчеството и специфично музикалните професионални умения и иновации, дори и когато в продукта се съдържат елементи от други изкуства или дейности (текст, графика, видео);
- тя е предимно инструментална музика; използването на човешки глас е допустимо, но фигурата на певеца като правило не я идентифицира пред слушателя, както често е например в поп или рокмузиката;
- в нея основното е творческата работа със звука (саунд-дизайн) и с пространствеността, което често предопределя и характера на самата композиция. Задължително е творческото (нестандартно) използване на електронна и компютърна техника в тези процеси, както и при окончателното оформление в студио или при живото изпълнение;
- в нея наред със синтезатори, електрически китари и компютърни генератори на музикален звук се използват често и традиционните акустични музикални източници: човешки глас, класически, фолклорни или екзотични инструменти, звуци от природата или урбанистичната среда, но те задължително минават през съществена обработка с компютри и други устройства на съвременната електроакустична техника.

Това именно е едно от основанията тази музика да се нарича електроакустична, за разлика от традиционната чисто електронна музика на авангардизма. Когато в процеса на самото композиране на тази музика се включат чисто компютърни средства за съчиняване (т.нар. алгоритмична музика), за автоматичен акомпанимент, за нюансиране на изпълнението или на пространствената локализация на звуците е по-правилно такава продукция да се нарича компютърна музика. Това са случаите, когато в авторството на съответното произведение съществен дял има и работата на компютърния програмист, наред с професионалните, чисто музикални творчески умения и усилия на самия композитор.

Електроакустичната музика още в своето начало вдъхнови и наложи едно романтично усещане, че новата технология, новото звучене и процес на композиране, ще се превърнат в съвсем ново водещо течение в музиката – с нова концепция, естетика и музикален израз. Не може да се каже дали това наистина се случи и ако е така в какъв размер. Но е очевидно, че втората половина на 20 век, характеризирана с изместването на аналоговите устройства от дигиталните системи, е ангажирана най-вече с проблеми от изчислителните технологии.

Все още има много за изследване относно как компютрите могат да ни помогнат да изразим и уловим своите музикални идеи, да експериментираме и да ги развиваме и

най-важното да ги организираме и да представим в завършен вид на изкуството. Този проблем може да бъде обобщен в един единствен въпрос: как да композираме музика с компютър?

Важно е да се изследват различните похвати и техники, които се развиват и прилагат за генериране на музиката с компютрите в академичната общност. Всяка една техника представя различен път и начин за мислене и усещане на музиката. Всеки един автор може да открие, че дадена техника е подходяща точно на неговото усещане за музика и да го използва, не само за да генерира нови музикални идеи, но също така да обработва предишни музикални творби, които е създал.

Фундаменталните въпроси свързани със създаването на музика със средствата на компютърните технологии са в анализа на приликите и разликите в начините за създаване на творби за инструменти или компютър и за творческото усещане при създаването на традиционна компютърна музика.

Тези въпроси са много уместни в днешно време, когато компютрите еднакво поддържат композирането на традиционни и експериментални форми на музиката. Но отговорите могат да дадат самите композитори, част от които създават музиката си използвайки традиционните похвати или използват индивидуални концепции, които прилагат единствено в собствения си специфичен музикален свят. Всеки, който се интересува от създаването на музика чрез помощта на компютъра може да открие полезна техника, близка до неговия стил или метод на композиране. В традицията на експерименталната музика има няколко принципа: опитвай, прослушвай и усъвършенствай. Основа на изследването са трите вида експериментална музика, чието изразно средство е електронното звукоизвличане.

1.1 КОМПЮТЪРНА МУЗИКА

Новото музикално пространство, в което живеем създаде нова музикална реалност, съставена от нови звуци, нови правила за свирене, за правене на звуци и нови изисквания при слушане.

Музиката е комплексна и абстрактна територия и е наследствено субективна. Тази субективност се изразява в множество различни концепции и подходи. Традиционната музикална теория, която се развива чрез вокалната и инструментална музика от няколко столетия насам, осигури много концепции, които формираха основните области от музикалното мислене и усещане, като нотирането и мелодичната

фразировка, мелодичният мотив, музикалната структура; динамиката, водещият вокал, контрапункта, хармонията и формата. По-новите музикални форми вече представиха нови начини за поглед над музиката.

Има две основни понятия свързани с музиката и те са, че музиката е съставена от звуци, организирани в пространството и времето. В този смисъл пространството основно се асоциира с вертикалните връзки между звуците, докато времето се асоциира с хоризонталните връзки. Второто понятие е че музикалните композиции притежават абстрактни структури. Тези понятия са важни за компютърната музика, защото те са адресирани към две отделни области на нашия музикален интелект: областта на абстрактната субективност (музикално композиране и артистично вдъхновение) и областта на абстрактната обективност (логически операции и математически модели).

Колкото повече се развива компютърната технология и софтуер, толкова повече предизвикателства ще застават пред компютърното музициране и импровизирането в различни жанрове и стилове.

Развитието на компютърните технологии исторически се развива паралелно с развитието на електронното музикално студио и синтезатора⁶. Това развитие не би било възможно без развитието на музикалния програмен език и метод за директен синтез⁷ на дигиталния сигнал в звук.

Електроакустичната музика от 60-те и 70-те години, направи възможно за първи път използването на аналогови устройства и компютри, окуражи композиторите да мислят различно, не само относно музиката и процеса на композиране, но също така относно усещането на музикалното преживяване. Електроакустичната музика се изгражда върху идеи, произлизащи от музикалния сюрреализъм и представя музикалната комплексност далеч над способността на човешкото изпълнение. На другата крайност е *Musique Concrete*, която насочва композиторите да откриват тембри, текстури, времеви потоци, преходи, фази и звукови морфеми. Накратко, ясно е че различните хора усещат своята музика по различни начини. Някои от тези начини са основни, а други израстват и се доразвиват в университетите, като експериментални средища. Съществуват много индивидуални начини, които се прилагат само към музикалния свят на особени и специфични композиции. Усъвършенстването на

⁶ Синтезаторът е апаратура с помощта на която се създават синтетични електронни звуци, електронна и компютърна музика.

⁷ Думата синтез е от гръцки произход и означава сумиране, свързване, съставяне. Синтезът е понятие с обща валидност, под което се разбира подходящо събиране на отделни, добре конкретизирани компоненти за постигане на определена цел. /5, с.63/

компютърната технология доведе до коренно различен подход в композирането на музика. Условно наречена «компютърна», тази музика има много подзаглавия, които в своята същност прецизират понятието. За да направим изследване на смисъла при композиране на фрактална музика ще поясним същността на компютърната музика.

1.1.1 КОМПЮТЪРНА МУЗИКА - ОПРЕДЕЛЕНИЕ

Компютърната музика е музика, генерирана или композирана с помощта на компютъра. Също така се отнася към област на проучване, която изследва както теорията, така и приложението на новите приложения в областта на музиката, звуковия дизайн и разпространение, акустиката, звуковия синтез, дигиталната обработка на звука и психоакустиката. Областта на компютърната област може да насочи своите основополагащи принципи обратно към началото на електронната музика и първите експерименти и иновации с електронни инструменти и техно музика в края на 20 век, за да иззчертае пътя на развитие и очакванията за бъдещото на компютърната музика.

С появата на нови програми компютърното моделиране започва да дава тласък за създаване на компютърна музика. Известни са четири вида съвместно музициране на изпълнителя с компютър: компютърът играе ролята на музикален инструмент; компютърът като интерпретатор; компютърът импровизира, като създава музикални фрази и променя параметрите на чути мелодии, като отговор на изпълнението от живи изпълнители, използвайки заложи в него програми; компютърът сам създава собствена музика на базата на серия от програмни алгоритми.

Терминът компютърна музика в наши дни има два различни смисъла. В академичните среди този термин преди всичко показва използването на формални алгоритми в процеса на музикалната композиция, а другия смисъл се намира в популярната сфера, където това се разбира като електронна музика, която използва компютърна техника [20]. Така музика често се определя и като алгоритмична. За еднозначния запис на алгоритми се използват формални алгоритмични езици, чрез които се строят езици за програмиране, в т.ч. и музикални езици (*Music 4, Sound, Supercollider, MAX/MSP*). [35]

Много от творбите за компютърна музика са написани в тясна връзка между музикалната теория и математиката. Първата компютърна музика в света е генерирана

в Австралия от програмиста Джеоф Хил (*Geoff Hill*), на CSIRAC компютър, който е конструиран от Тревър Пиърси (*Trevor Pearcey*) и Мастън Беърд (*Maston Beard*).

През 1951 г. е направен първия запис на компютърна музика от Би Би Си. Първият достъпен компютър *Ferranti Mark I* изсвирва песните “*Baa Baa Black Sheep*” , “*God save the King*” и “*In the Mood*”.

През 1957 г., Едгард Варезе (*Edgard Varese*) работи над творбата си *Poeme electronique* на *RCA Mark I* - електронен музикален синтезатор в Принстън, Ню Джърси. А в лабораториите Bell Макс Матюс (*Max Mathews*) успява да синтезира чрез компютър няколко музикални тона. Резултатът от това е представлявал кратък монофоничен откъс, с продължение от 17 секунди, но всъщност това е била първата програма написана да генерира звук, директно от компютъра. Програмният език, който създава Матюс се нарича MUSIC I и е имал ограничение за един глас, една звукова вълна и не е имал контролери върху динамиката на звука. Това което е можела да прави програмата е било последование от височини на определени интервали, с определна сила, за определен интервал от време. Тази програма бива подобрена от няколко други версии. MUSIC II (1958) добавя четири гласа и концепцията на синтезатор, в който всички дигитални параметри за звука се извикват от дефинирани таблици. В MUSIC III (1960) Матюс въвежда няколко други концепции:

- Библиотеки с функции
- Генератор за синтез
- Оркестри
- Партитури

С появата през 1969 г. на програмата MUSIC V Матюс предоставя версия програмирана на FORTRAN. След това Матюс се насочва към разработването на GROOVE през 1970 г. (*Generated Real-time Output Operations on Voltage-controlled Equipment*), една компютърна система, която изобразява чрез потребителски интерфейс управлението на програмата в реално време.⁸

През 1968 г. Матюс и Рослър (*L.Rosler*) отново в лабораториите на Bell, разработват Graphic 1, една интерактивна компютърна система, която е можела да превежда образи, изрисувани на екрана, в синтезиран звук. Матюс написва: “Graphic 1 позволява на потребителя да вмъква картини, които благодарение на мощта на

⁸ Както обяснява самия Матюс: “Компютърният изпълнител не трябва да се стреми да дефинира цялостния звук в реално време. Това ще го направи компютъра, докато изпълнителят ще повлияе на начина, по който да се изсвирят нотите...”

компютъра, да могат да бъдат модифицирани, изтривани, копирани и съхранявани”. С разработката на тази система Матюс е представил концепцията на интерактивната композиция в реално време върху компютърния екран, с *cut u paste* възможности, години преди персоналният компютър да направи тези функционалности достъпни за потребителите. Леджарън Хилър (*Lejaren Hiller*) пръв изследва употребата на компютъра, като средство да се композира инструментална музика през 50-те години. Той работи с Робърт Бейкър (*Robert Baker*) и написва програмата MUSICOMP за компютър IBM 7094. Програмата е била важна поради това, че е разчитала на похвати, базирани на правила, за да се конструират откъси от музика от предефинирани променливи - нещо, което се случва десетилетия преди създаването на MIDI⁹ и персоналният компютър.

Янис Ксенакис (*Iannis Xenakis*) се е интересувал от музика, основана на теория на вероятностите и асоциирана с математически процеси. Той композира творби, много от които използват генерирането на огромни маси от звук. Неговата първа стохастична композиция е била изчислена ръчно.

1.1.1.1 КОМПЮТЪРНО КОМПОЗИРАНЕ (*COMPUTER ASSISTED COMPOSITION*) ИЛИ КОМПОЗИРАНЕ, ПОДПОМОГНАТО ОТ КОМПЮТЪР

Това е техниката или практиката за използване на компютър, като средство за композиране на музика, въпреки че музиката може да бъде изпълнена, както електронно, така и по традиционния начин, използвайки неелектронни инструменти, без употребата на компютър или електронно устройство. Две от най-важните страни на музиката, композирана с компютър са алгоритмичното композиране и спектралната музика.

1.1.1.2 КОМПЮТЪРНО-ПОДКРЕПЕНО АЛГОРИТМИЧНО КОМПОЗИРАНЕ (*COMPUTER-AIDED ALGORITHMIC COMPOSITION*)

СААС е осъществяването и употребата на техниките за алгоритмично композиране в софтуера. Този термин е постигнат от комбинацията на два етикета, всеки от които е твърде неясно формулиран, за да продължи да бъде използван. Етикетът "*computer-aided composition*" е недостатъчен да обясни спецификата на използването на генериращи алгоритми. Музика, създадена с нотация или някакъв

⁹ MIDI (Musical Instrument Digital Interface) направи възможен обменът на информация между музикалните инструменти по електронен път /6, 117/

софтуер лесно може да бъде сметната за компютърно-подпомогната композиция. Етикетът "алгоритмично композиране" е също така твърде обширен, особено за това не конкретизира употребата и на компютъра. Терминът компютърно-подкрепен (*computer-aided*), по-точен от термина компютърно-подпомогнат (*computer-assisted*), се употребява по същия начин, както и компютърно-подкрепен дизайн (*Computer-Aided Design*).

1.1.1.3 КОМПЮТЪРНО-ГЕНЕРИРАНА МУЗИКА (*COMPUTER-GENERATED MUSIC*)

Това е музика, композирана от или със силната подкрепа на компютъра. Въпреки че всяка музика, която използва компютри в своите композиции или реализации – е компютърно генерирана в известна степен, употребата на компютри сега е толкова разпространена (например, при редактирането на поп песни), че фразата компютърно-генерирана музика общо се използва, за да обозначи вид музика, която не би могла да бъде създадена без използването на компютър.

Можем да разграничим две групи от компютърно-генерирана музика: музика, в която компютър генерира нотите, които могат да бъдат изпълнени от музиканти и музика, която едновременно е създадена и изпълнена от компютъра.

1.1.2 ИСТОРИЧЕСКИ ОБЗОР ВЪРХУ РАЗВИТИЕТО НА КОМПЮТЪРНАТА МУЗИКА

В съвременната история на компютърната музика за първи път в света се генерира компютърна музика в Австралия от програмиста Джеоф Хил (*Geoff Hill*) на CSIRAC компютър. Впоследствие, Леджарън Хилър (*Lejaren Hiller*) използва през 1950 г. компютър, за да композира творби, които след това се изпълняват от традиционни музиканти. Един от първите композитори, които почват да създават музика с компютър е Янис Ксенакис (*Iannis Xenakis*). Той написва програми на езика FORTRAN, който генерира числови данни, които той пренаписва и възпроизвежда в ноти, които да бъдат изсвирени от традиционни музикални инструменти. Един пример е ST/48 през 1962. Въпреки че Ксенакис е можел да композира хубава музика по този начин, и то ръчно, дълбочината на изчисленията, необходима да трансформира вероятната математика в музикална нотация, е била най-добре извършвана от компютрите.

Има различни технологии за създаване на компютърна музика. Vocoder технологията е важно откритие в този ранен етап на развитие. От своя страна MIDI технологията позволява на персоналните компютри да си комуникират със синетазторите чрез стандартизиран интерфейс, който разшири употребата на компютърните технологии в музиката.

В световен мащаб заслуга за съществуването на компютърната музика имат много организации и институции, посветили работата си на изследването на компютърната и електронна музика, включително ICMC (*International Computer Music Association*) – Международна асоциация за компютърна музика, IRCAM, GRAME, SEAMUS (*Society for Electro Acoustic Music in the United States*) – Общество за електроакустична музика в Съединените щати, както и много университети в целия свят .

1.1.3 МУЗИКА И НОВИ ТЕХНОЛОГИИ

В по-неотдавнашните столетия, музиката е държана отделно и настрана от нейните детерминистични основи. Музикантите често се примиряват със слабо дефинираните тълкувания на феномени, като "вдъхновение" и "гениален", всеки път, когато е била провокирана теоретичната реабилитация за музиката. Успоредно с това математиката продължава да преследва и да разработва свои цели от точните логически изрази, достигайки различни постижения, адекватни на Нютоновата механика и диференциалното смятане.

В днешно време обществото се връща обратно към хипотезите на Питагор и ресинтезира някои параметри. Вследствие на научните проучвания и изследвания, обществото вече използва модерни възгледи, свързани с използването на нови електронни инструменти или създава мултимедийни продукти на компютъра си, или индиректно слуша музика от цифров носител. [36]

От друга страна уважаваните математици тренират своята интуиция със "забавното" проучване над компютърното визуализиране, а "вдъхновените гении" правят блестящи проучвания в областта на синтезаторите, секвенсирите и дигиталните процесори на сигнали.

Днешните съюзи между музиката и математика се характеризират със създаването на музика, използвайки математически базирани инструменти.

1.1.3.1 КОМПЮТЪРНО-ГЕНЕРИРАНИ КОМПОЗИЦИИ ЗА ИЗПЪЛНЕНИЕ ОТ МУЗИКАНТИ

Компютрите също така са използвани в опита да имитират музиката на великите композитори от миналото, като Моцарт. Един настоящ представител на тази техника е Дейвид Коуп (*David Cope*). Той написва компютърни програми, които анализират творбите на други композитори, за да представя нови творби в сходен на техния стил. Той използва тези програми за по-голям ефект с композитори, като Бах и Моцарт (неговата програма *Experiments in Musical Intelligence* е най-известната за създаването на "*Mozart's 42nd Symphony*"), но също така и за създаването на негови собствени композиции, комбиниращи неговите собствени творби с тези на компютъра .

Някои критици твърдят, че компютрите са неспособни да създават музика със същото качество и ниво, като това на великите класици, макар че други биха попитали дали това е най-важното при създаването на музика по този начин.

1.1.3.2 МУЗИКА, КОМПОЗИРАНА И ИЗПЪЛНЕНА ОТ КОМПЮТРИ

На по-късен етап, композитори, като Готфрид Майкъл Кьониг (*Gottfried Michael Koenig*) използва компютъра да генерира звуците на композициите си, както при нотната партитура. От самото начало Кьониг създава алгоритмични композиционни програми, които са подобни на създадената от Ксенакис по това, че те транслират изчислението на математическите уравнения в музикални партитури, които да могат да бъдат изпълнени от музиканти. [40] Неговите програми Project 1 и Project 2 са примери от този вид софтуер. Впоследствие, той разширява същите принципи в сферата на синтезирането, давайки възможност на компютъра да създава директно музика. SSP е пример за програма, която изпълнява тази функция. Всички тези програми са представени от Кьониг в Института по сонология в Утрехт, Холандия, през 1970 г.

Такива способности, като тези използвани от Кьониг и Ксенакис се използват и до днес. Още от създаването на MIDI системата през 1980 г., някои композитори работят с програми, които преобразуват MIDI нотите към алгоритъм и тогава могат както да възпроизвеждат музиката чрез звуковата карта на компютъра, както и да записват аудио файл за други програми, които да го просвирват.

Някои от тези прости програми се базират върху фракталната геометрия¹⁰ и могат да изчертават миди нотите към конкретни фрактали или фрактални уравнения. Въпреки че подобни програми са широко достъпни и понякога се считат за електронни играчки за хора, които не са занимават с музика, някои професионални музиканти са им

¹⁰ Виж речника

обърнали специално внимание. Музиката, която се създава с такива програми може да наподобява понякога шум или може да звучи съвсем приятно и мелодично. Както при алгоритмичната музика, така и при алгоритмичното изкуство като цяло, всичко зависи от начина, по който параметрите се изчертават към аспектите на тези уравнения, а не от самите уравнения. Поради това, например, същото уравнение може да бъде използвано да създава както лирична и мелодична музика в стила на 19 век, така и фантастични дисонантни какафонии, напомнящи авангардната музика на 50-те и 60-те години на 20 век.

Други програми могат да изчертават математическите формули и константи, които да възпроизвеждат поредици от ноти. По този начин, едно ирационално число може да създаде безкрайно последование от ноти, където всяка нота е десетичен израз на това число. Това последование може да се превърне в композиция или просто да стане основата за бъдещи разработки.

Операции като тези и дори по-сложни операции могат да бъдат изпълнени чрез компютърни музикални програмни езици, като *Max/MSP*, *SuperCollider*, *Csound*, *Pure Data (Pd)*, *Keykit* и *ChucK*. Тези програми сега лесно се изпълняват на повечето персонални компютри и често са способни да изпълняват много по-сложни функции от онези, които са били наложителни за най-мощните компютри няколко десетилетия по-рано.

Съществуват програми, които генерират човешко-звучащи ("human-sounding") мелодии, използвайки обширна база данни от фрази. Един пример за това е *Band-in-a-Box*, който е способен да създава джаз, блус и рок инструментални солота, почти без никаква намеса от потребителя.

Друг кибернетичен опит за компютърно музициране е използването на специализиран хардуер, който открива външни стимули, които след това се преобразуват от компютъра, за да реализира изпълнението. Примери за такъв стил компютърна музика могат да бъдат открити от средата на 80-те години в творби на Дейвид Рокибай (*David Rokeby*) „*Very Nervous System*”, където емоциите и жестовете на публиката или на изпълнителя, се транслират към MIDI сегменти. Компютърно контролираната музика също така може да бъде открита в творбите на канадския композитор Удо Касеметс (*Udo Kasemets*) (1919, обработената пиеса *Music Errata*, използваща модел на електрически влак, за да направи самосвал, пълен с камъни, които да бъдат поставени върху ударен инструмент, свързан към Аналого/Дигитален преобразувател, преобразуващ ударите от камъните към нотен текст (изпълнено в

Торонто от пианиста Гордън Монахън през 1987г.). В по-късните творби, компютърът генерира звукови стъбла от касетъчни, повтарящи се звукови семпли, чрез синусоидни генератори. [36]

1.1.4 СИНТЕЗ НА ЗВУК ЗА КОМПЮТЪРНА КОМПОЗИЦИЯ

Звукът се произвежда от традиционни (акустични) музикални инструменти и електронни музикални инструменти (напр. синтезатор). Звукът, който се получава от по електронен път носи наименованието синтетичен електронен звук. Електронните музикални инструменти синтезират нов електро-акустичен аналог на традиционния акустичен звук. Използването на методи за дискретизация на аналоговия електрически сигнал, чрез аналогово-цифрови преобразуватели, реализира цифровия синтез на звука.¹¹ Цифровият звуков синтез може да бъде модифициран и управляван с помощта на микропроцесорна система. «Синтезът на звук с цифрови средства се свежда до генериране на поредица от дискрети (0 и 1), която представя звуковия сигнал».¹² Същевременно цифровият сигнал се нарича електрически сигнал, чието напрежение във всеки определен момент има стойност, принадлежаща към група дискретни нива.¹³

1.1.4.1 АУДИО ПРОГРАМЕН ЕЗИК (AUDIO PROGRAMMING LANGUAGE)

Това е програмен език, специфично насочен към създаването на музика или синтезирането на звук.

ChucK е аудио програмен език за синтез на звука в реално време, композиране и изпълнение, който работи върху операционни платформи на Mac OSX, Linux и Windows. Изработен е за по-добра четливост и гъвкавост за програмиста, за сметка на други съображения като суровото изпълнение. Поддържа равностойни и множество едновременни нива на динамичен контрол. Друга ключова черта е способността да допълва, премахва и модифицира кода в процеса на работа – (*code on the fly*), докато програмата работи, без да се спира или рестартира. Има висока прецизност на времевия модел. Предлага на композиторите и изследователите мощно и гъвкаво средство за програмиране, за изграждане и експериментиране със сложни аудио синтезиращи програми, както и интерактивен контрол в реално време.

¹¹ Този процес днес се познава като семплиране (обработка). /3, с.8/

¹² Лазоров, С. Електронна музика и синтезатори. С., Техника, 1986, с.50

¹³ Следователно напрежението не се променя непрекъснато (за разлика от аналоговия сигнал). /3, с.70/

1.1.4.2 ЖИВО ПРОГРАМИРАНЕ - *LIVE CODING*

Понякога познато като интерактивно (*interactive*) програмиране, *on-the-fly* програмиране е името дадено за процеса на писане на софтуер в реално време, като част от дадено изпълнение. В исторически аспект, тази техника е позната още от времето, когато компютрите са се използвали да показват ранното компютърно изкуство, но неотдавна бе изследвана като по-строга алтернатива при лаптопите на DJ, които кодират на живо, като така се усеща осезаемата липса на обаянието и привлекателността на живото музикално изпълнение.

Общо взето, тези практически нива са по-обобщени подходи: един е интерактивното програмиране и писането на програми, в момента, в който те се изпълняват на живо. Традиционно повечето компютърни музикални програми клонят към старите похвати на писане /компилиране/ изпълнение, които се появяват, когато компютрите са били много по-слаби откъм мощност. Този похват отключи нивото на иновации при програмирането за хора, чиито програмиращи способности са по-скромни. Някои програми имат постепенно интегрирани контролери в реално време (например, драйвер за MIDI-софтуерен синтезис и параметричен контрол). До неотдавна, музикантите и композиторите рядко имаха способността за модификация в реално време на програмния код. Тази отличителна черта до известна степен изпъква чрез програмни езици, като *Chuck* и *SuperCollider*. Музиканти, които не са имали възможността да придобият технически познания, за да възприемат бързото нарастване на възможностите на повечето музикални програми, сега могат да експериментират с кодовете в дадена обстановка, които вече са много по-интуитивни в музикално отношение (макар че все още са повече технически, отколкото музикални).

1.2 АЛГОРИТМИЧНА МУЗИКА

Творчеството най-вече е процес на спонтанност, когато се дава път на вдъхновението, усещането, таланта. Много често музиката наподобява рекурсивен алгоритъм, който се опитва да достигне някои оптимални функции, с добри качества и следователно е по-лесно осъществимо чрез една компютърна програма. Заради това

дефинираме ясно целта на една алгоритмично композиционна система: да пресъздаде творческата методология на композитора, когато композиторът създава музика не на базата на вдъхновението си, а на упорита работа. Резултатът е пригодена система за този специфичен тип композитор; ако друг творец е използвал системата, за да създаде музика, тогава той композира музиката на предишния композитор, не своя собствена.

1.2.1 АЛГОРИТМИЧНА МУЗИКА - ОПРЕДЕЛЕНИЕ

Алгоритмичното композиране е техниката да се използват алгоритми, за да се създава музика. Алгоритмите или поне редицата от правила са се използвали за композиране на музика от векове. Терминът изразява употребата на формални процедури за създаване на музика без човешка намеса, преминаващи през случайни процедури или употреба на компютри. Има радикална разлика между композитори, които използват стохастични процедури да композират музика и тези, които използват програми, които продуцират определени резултати, вследствие на строго фиксирани алгоритми. Алгоритмичната музика е музика, създадена от някакъв автоматизиран процес. Доколкото сме виждали и слушали примери от алгоритмична компютърна музика, разбираме, че алгоритмичната музика не е просто някаква подкатегория от компютърната музика. Изчисляването не е задължителна черта на един алгоритъм. Има множество примери на автоматизирано създаване на музика чрез физически средства. Основното е че алгоритмичната музика не трябва да се бърка с автоматизираното музикално изпълнение. [42] Алгоритмичната музика е свързана главно с композирането. [35] Исторически и в настоящето се открояват три основни подхода за алгоритмично композиране с компютър: алгоритми за звуков синтез; алгоритми за композиционна структура и алгоритми за корелация на звуковия синтез със структурата. Звуково синтезиращите алгоритми са били използвани по много и различни начини - от изчислението на сложни звукови вълни изграждащи звука, до еволюцията в усъвършенстването на тембъра.¹⁴

Самата алгоритмична музика може да се типологизира в две категории: алгоритми основани на правила от традиционната «класическа» или популярна музика; алгоритми,

¹⁴ A.Kurepa and R.Waschka, "Using Fractals in Timbre Construction: an Exploratory Study," in Proceedings of the 1989 International Computer Music Conference (San Francisco:International Computer Music Association, 1989), 332-335

които имитират направления от модерната «сериозна» музика на XX век (алеаторика¹⁵, додекафония¹⁶, стохастична музика) или директно използват някои сходни подходи от арсенала на математиката¹⁷ (комбинаторика, генератори за случайни числа и др.). [4]

1.2.2 ИСТОРИЧЕСКО ИЗСЛЕДВАНЕ ВЪРХУ РАЗВИТИЕТО НА АЛГОРИТМИЧНАТА МУЗИКА

Както бе посочено в първа глава според Флавиус Касиодорус “Музиката е дисциплината, която има отношение към числата и тяхната връзка към онези неща, които се откриват в звуците”.

Във днешни времена, музиката и математиката – едното изкуство, другото наука, се свързват заедно обединени от един общ логически фактор – откриват истинската си връзка чрез компютъра, чрез чието съдействие двете дисциплини съзнателно и практически се интегрират. Съвременната поява на софтуерните програми, създадени за алгоритмично композиране, надстрои идеята на античния човек да търси все по-мощни инструменти за генериране на музика.

Терминът алгоритъм е бил взиман от областта на математиката и компютърните науки. Необходимо е да се изясни основното недоразумение, което придружава термина “алгоритмично композиране”, за да се разреши нарастващото чувство за двусмисленост.

Думата алгоритъм се използва повече от хилядолетие. За да се изясни неговата употреба във връзка с музикалното композиране, е нужно да се обърне внимание на етимологията на думата. Думата алгоритъм се появява през 9 век от работата на арабския математик и писател *Abu ‘Abdullah Muhammad ibn Musa of Khwarizm*. Неговата най-значителна книга, със заглавие “*al-Kitab al-mukhtasar fi hisab al-jabr wa’l-muqabala*”, което в превод означава «Правила на реинтеграция и редукция», става основата за стандартизация на арабските числа в европейската математика. Наистина световната алгебра е средновековна латинизация на Арабския свят *al-jabr*, което означава буквално “редукцията”. Латинската основа на думата алгоритъм определено е наследена от тази родина на математиката, *al-Khwarizm (The Khwarizm)* и термина “*algorismus*”. Английската форма “*algorism*” е използвана, за да означава “Арабската или десетичната система на числата”. Тази архаична дума се появява буквално преди

¹⁵ Виж речника

¹⁶ Виж речника

¹⁷ Кавалджиов, Л. Съвременна музикална терминология. Българско музикознание, 1994. с.36

повече от 1200 години и е необходим само един малък филологически аргумент, който да заключи, че думата е била предшественик на съвременния термин алгоритъм.

Един алгоритъм в съвременните термини се характеризира от следните възможности:

1. Един алгоритъм съдържа определена последователност от действия;
2. Последователността от действия има начално и крайно действие;
3. Действието на алгоритъма се прекратява, както когато намери решение на проблема, така и когато стигне до извода, че проблемът е нерешим.

В по-срещаните определения, алгоритъм е процес, който решава даден проблем стъпка по стъпка, като едновременно използва разпределение, рекурсия или разклонение. Решението трябва да бъде достигнато в определен брой стъпки. В приложението им в музиката, алгоритмите могат да бъдат считани, като процедури, които проверяват дали е подходящ потенциалния музикален материал в даденото съдържание. Височината, трайността, динамиката и други звукови и структурални константи могат да бъдат избирани, съобразно от група въпроси и отговори. Нужно е да се каже, че компютъра е задължителен инструмент за обединяването на алгоритмичните процеси в музикалната композиция. Употребата на алгоритмите, водена както от количеството, така и от качеството им, е идеално поставена задача за компютъра. Най-добре използван в този си капацитет е компютъра, като устройство спестяващо излишен труд, даващо свобода на композиторите и неангажиращ ги с изчислителната част от този процес.

Компютърът е един инструмент за реализирането на абстрактни конструкции, които впоследствие да бъдат приложени в музиката, съобразявайки се с креативното мислене и вдъхновение на композитора. Използван благоразумно и съобразно с естествените параметри на неговите умения, компютърът позволява на композитора да експериментира с музикалните възможности на различни алгоритмични процедури и да оценява тяхното потенциално развитие в квалифицирани композиционни системи. [30] Компютърът в алгоритмичната музика, е най-подходящо приложим като композиционен инструмент, отколкото само като едно музикално решение.

В по-младата историята на алгоритмичното композиране с компютри, изникват три основни похвата: алгоритми за звуков синтез; алгоритми за композиционна структура; алгоритми за връзка на звуковия синтез със структурата. [43]

Звуково синтезиращите алгоритми са използвани по различни начини – от генерирането на сложни звукови вълни до еволюцията на тембровото усъвършенстване. Някои от алгоритмичните програми и композиции определят само партитурната информация. Партитурната информация включва височина, трайност, динамика, написани за акустични и електронни инструменти: например има случаи, в които композиторът използва компютърната програма да генерира партитурата, докато инструменталната селекция се предопределя както за електронен оркестър или за реализация от акустични инструменти.[41]

Други алгоритмични програми определят както партитурата, така и електронния звуков синтез. В този случай, програмата се използва не само да генерира партитурата, но и да се използват електронни тембри в изпълнението.

Една от основните концепции заложили в алгоритмичното композиране е кодово-графичното преобразуване (*mapping*). Кодово-графичното преобразуване се състои от създаването на директна връзка между алгоритмичния изходен резултат и музикалните параметри. Кодово-графичното преобразуване също така може да бъде разбрано като превод на алгоритмичния музикален материал в музика. Как ще бъде представено кодово-графичното преобразуване изцяло зависи от композитора. Трябва да се подчертае, че има много начини да се преобразува алгоритмичния числов резултат към музикални параметри. Един и същ алгоритъм ще възпроизведе съвсем различна музика според приложението на различно кодово-графично преобразуване. В допълнение към височината, кодово-графичното преобразуване може да бъде приложено към всеки един музикален параметър (ритъм, динамика), както и към други музикални събития, като мотивите или дори напълно завършени музикални фрази.

Едни от най-използваните категории на алгоритми, които могат да бъдат приложени към музикалното композиране са:

- Стохастични (*Stochastic*) процеси (вероятностни функции, вериги на Марков);
- Рекурсивни (хаос, фрактали, нелинейни уравнения, числова теория);
- Базирани на правила (*L-system*);
- Генетични алгоритми.

Всеки един от тези процеси има свои собствени характеристики и музикалния материал, който генерират варира съобразно това.

Стохастичните процеси, които включват употребата на вероятностни функции, са първите, които са изследвани от Янис Ксенакис през 1950 г., по време когато авангардната музика е доминирала с наследената от Шьонберг 12-тонова система.¹⁸ Желанието да се намерят нови пътища в музикалното композиране кара Ксенакис да формулира една нова парадигма за музиката, която той нарича стохастична (*Stochastic*) музика.[78]

Употребата на тези нови техники е изисквала компютрите да бъдат специално програмирани.

Преди десетилетия когато употребата на компютри е била много ограничена и е липсвал софтуер, композиторите е трябвало да пишат сами програмите, с които да композират или са работили съвместно с програмисти. Това предполага дълбоко познание на алгоритмичната математическа структура, както и много добри програмистки умения (по това време се е програмирало на езици на много ниско ниво като Асемблер или машинен код). Не е изненадващо, че композиторите, които са използвали тези техники, също така са имали познания в областта на математиката и компютрите (Янис Ксенакис е бил архитект и математик; Леджарън Хилър – е бил химик).

Днес, благодарение на развитието на технологиите и на обектно-ориентираните програмни езици, някои от които са структурирани специално за музикално композиране¹⁹, композиторите вече могат да имплементират алгоритмични процедури по много по-интуитивен начин. Понеже тези програмиращи среди са специално структурирани за композиране на музика, композиторът спестява много време от това, че не участва в кодиращия процес. Дължината на кода в един алгоритмичен процес включен в CSound типично е от 10 до 15 пъти по-къс, отколкото ако е програмиран на C.

Алгоритмичната музика в основата си използва Питагоровата концепция за вездесъщата същност на числата, възникнала от изследването му над музикалните интервали и има за цел да наблюдава “орфическия катарзис”, в съответствие с вярването, че музиката пречиства душата, както медицината пречиства тялото.

Теорията чрез която Питагор открива тези съотношения е чрез измерване дължината, съответстваща на тях в монохорда.²⁰ Методът открит от гърците за

¹⁸ Xenakis, *Formalized Music*, 14-15

¹⁹ Единият от тези програмни езици е CSound, а другият MAX

²⁰ Xenakis, *Formalized Music*, 202

установяване на няколко интервала, базирани на Питагоровата скала, се базира на аритметични операции с пропорции. Например, вадейки квинта от октава получаваме кварта (2:1 / 3:2 = 4:3); вадейки кварта от квинта получаваме цял тон (3:2 / 4:3 = 9:8); този процес се повтаря, създавайки по-малки и по-малки интервали: умалена терца (32:27) и Питагоровия интервал (531441:524288), който има честотна разлика от 23.5 цента (темперираният полутон е равен на 100 цента). Тези съотношения могат да бъдат също така представени чрез аритметика и хармонични средства. Средното аритметично на две числа е равно на сумата от числата, разделена на две, т.е.

$$a = \frac{x + y}{2}.$$

Ако a е средното аритметично на x и y , приемайки че $x > y$, то $x - a = a - y$. Средното хармонично на две числа е равно на реципрочната стойност на средното аритметично на реципрочните им стойности, т.е.

$$h = \frac{2}{\frac{1}{x} + \frac{1}{y}} = \frac{2xy}{x + y}.$$

Ако h е средното хармонично на x и y , приемайки че $x > y$, то $(h - y)/y = (x - h)/x$.

Друга важна пропорция използвана в античността [34] е била средното геометрично g на две числа x и y , което е равно на корен квадратен от произведението им, т.е.

$$g = \sqrt{xy}.$$

Ако g е средното геометрично на x и y , то $(x / g) = (g / y)$.

Аристид Куинтилианус (*Aristides Quintilianus*), един от първите музикални теоретици, изследва връзките между средното аритметично, средното геометрично и средното хармонично на две числа.²¹

Откриването на музикалните съотношения от Питагор достига върха си в сложната гръцка музикална теория, на която Аристоксенос (*Aristoxenus*- роден 365 г. пр.Хр.) е бил първият и най-важен теоретик.²² По-късните музикални теоретици са

²¹ “Аритметично и хармонично значение”

²² Xenakis, *Formalized Music*, 183-189

биват повлияни от това математическо обяснение на музиката от гърците. Тези теоретици и техните музикални трактати са Куинтилианус (*Quintilianus (De musica)*), Птолемей (*Ptolemy (Harmonics)*), Боеций (*Boethius (De Institutione musica)*), Гаудентиус (*Gaudentius (Harmonic Introduction)*), Хуцбалд (*Hucbald (De harmonica institutione)*), Царлино (*Zarlino (Institutioni harmoniche)*), Фукс (*Fux (Gradus ad Parnassum)*), Рамо (*Rameau (Traite de l'harmonie)*). [78]

1.2.2.1 СРЕДНИТЕ ВЕКОВЕ

Най-ранните известни примери за алгоритмични методи, приложени към музикални композиции и първия пример за кодово-графично преобразуване, датира от 11 век от италианския композитор и теоретик Гвидо Д'Арецо.²³ Неговият метод се състои от създаване на връзка между гласните в един текст и поредица от височини. [49]

В синтезирания алгоритмичен модел височините на един стандартен диапазон от две октави са представяни слогово по следния начин:

Г А В С D E F G a b c d e f g a

Където Г съответства на G под С от първа октава. Следващите три итерации на гласните “a e i o u” са заместени под височините, изглеждащи по този начин:

Г А В С D E F G a b c d e f g a
a e i o u a e i o u a e i o u a

След това гласните от текста биват изваждани. Накрая, използвайки табличката от височинни съотношения по-горе, композиторът е генерирал мелодия. Понеже всяка гласна може да бъде преобразувана към три различни височини (с изключение на гласната а, която има четири кореспонденции), една поредица от n гласни може да генерира поне 3^n различни мелодии (този брой може да бъде дори по-голям, когато гласната има 4 възможни преобразувания). Изборът измежду тези три възможни

²³ Loy, “Composing with computers”, 303

преобразувания за гласна (четири в случаите ако е а) се взима от композитора. Това позволява на композитора да прави такъв избор, че мелодичните контури да се приспособят към стилистичните правила на времето. Въпреки че този метод в основата си е алгоритмичен, неговата определеност е по някакъв начин неясна от експоненциалното нарастване на възможните мелодии като брой гласни в нарастващия преобразуван текст (за текст с 15 гласни и само няколко думи, има поне 3^{15} над 14 милиона възможни мелодии).

Изоритмични мотети²⁴ е музикална техника, която аранжира фиксирани образци от височини, с повтарящи се ритмични образци. Тази техника датира от 14-ти и 15-ти век и е друг пример, в който един математически процес се смесва с музикалната композиция. Изоритъм е термин, който е бил въведен от Фридрих Людвиг през 1904 г., означава повторение на ритмични и мелодични образци през вокалната партия (най-вече партията на тенора), въпреки че в късните години на 14-ти и ранните години на 15 век много композиции са били изоритмични във всички гласове, което се нарича панизоритмик – *panisorhythmic*. Ритмичният образец се е наричал талеа (*talea*), а мелодичният образец – цвят (*color*). Двата образа могат да имат различна дължина, така че успешните повторения на *talea* могат да се случат с различни височини. Ако една *talea* се състои да кажем от 10 трайности, а цвета се състои от 6 височини, ще имаме 30 (5x3x2) възможни пермутации на двата образа, докато не започнат да се повтарят отново. В панизоритмичните мотети, броя на възможните пермутации може да нараства астрономически. Естествено, от всички възможни комбинации, само една група от тях са били позволени от хармоничните правила на съответното време.

Композитори, които са използвали тази техника, включително Витри, който пръв е използвал тази техника, са Чиконя, Макаут, Дънстейнбъл, Дуфей (*Ciconia, Machaut, Dustanble, Dufay*). След 15-то столетие, изоритъмът бавно изчезва и се налага да изчакаме до 20-ти век, за да се открият сходни на него практики.

Друга процедура развита по време на Средните векове, която включва кодово-графично преобразуване на немускален материал към музика е т.нар. *soggeto cavato*. *Soggeto cavato* е термин, който бива въведен от Царлино и се състои от правене на връзка 1:1 между индивидуални думи в текст към музика. За пример, месата на Жоскен Де Пре, посветена на *Hercules Dux Ferrarie*, предоставя темата си от това посвещение, като използва 6-те солмизиращи срички или имената на тоновете от гамата (*ut, re,*

²⁴ Isorhythmic (от гръцки – същия ритъм)

mi, fa, sol, la), както следва: *re, ut, re, ut, re, fa, mi*, *re* съответстващи на гласните *e-u-e-u-e-a-i-e*, пораждащи тоновете : *d c d c d f e d*. *Soggetto cavato* е техника, която се използва от композиторите през цялата история на музиката.[79]

Добре познати примери включват творба на Бах, в която той използва собственото си име (В А С Н : В – flat – А – С – В неалтеровано) във финала на “Изкуството на фугата”; Вариациите за пиано на Шуман ABEGG (А - В – flat – Е – G – G); буквата В в немския език съответства на В flat (си бемол), а Н на В неалтеровано (чисто си). Тази процедура може да бъде приложена към всички букви от азбуката, не само към онези, които подсказват височините на тоновете. Например, Равел в неговия *Menuet sur le nome de Haydn* преобразува “*Haydn*” като В – А – D – D – G, преобразувайки “у” към D и “n” към G, както следва:

A B C D E F G H I K L M **N** O P Q R S T U V W X **Y** Z

A B C D E F **G** A B C D E F G A B C **D** E

Част от гореизброените алгоритмични методи се използват и днес в създаването на алгоритмичен модел.

1.2.2.2 КЛАСИЧЕСКИ ПЕРИОД

Много системи, създадени за генериране на музикални композиции съществуват формално преди появата на компютрите. Интересен пример за приложение на математически процеси в музиката е системата *Musikalisches Würfelspiel* на Моцарт. Тази система използва хвърлянето на зарове, за да избира на случаен признак измерванията от голяма колекция, състояща се от малки фрази. Тези фрази се комбинират, за да създадат музикални пиеси, които впоследствие могат да бъдат изпълнени от музиканти. Въпреки че тези творби не са композирани с компютър, може да се счита, че в тях Моцарт използва основни форми и алгоритмични техники, които сега композиторите често използват с появата на електронните двоични дигитални компютъри.

Идеята на тази музикална игра е заедно да се отрежат и залепят вече композирани тактове, за да се създаде менует и трио. Тази техника е открита от Кирнбъргър през

1757 г.²⁵ В късния барок тази техника е общоприета практика сред композиторите, дори и измежду тези с голям авторитет, да използват компилации от прогресии, каденци, мотиви и т.н. в техните творби, понякога дори като източник на вдъхновение, понякога и като конкретна музика. Тази техника е позната като *ars inveniendi*. [49]

Моцарт е използвал таблица от 176 възможни такта за 16-тактов Менует (11 2 зарови различни комбинации по 16 такта) и 96 за трио секцията (6x16). Структурата на композицията е била определяна на случаен принцип – два зара са били хвърляни, за да определят тактовете за менуета, един за тактовете на триото. Резултатът от хвърлените зарове е бил проверяван в таблицата на тактовете, за да се определи кой да бъде изсвирен.

Инструкциите за *Musikalisches Würfelspiel* фигурират на страниците със скечове в Националната библиотека на Париж, датиращи от 1787 г.²⁶ Композиционните процедури са както следва:

- хвърляне на 2 зара за 1 такт на менуета;
- сравняване на числото на зарчето (което е било необходимо да е число от 2 до 12), за да съответства на числото от таблицата на 176-те композирани тактове;
- повтаряне на горните две процедури, докато не се определят всички 16 такта.

Триото се е формирало аналогично на горните процедури, с тази разлика, че се е използвало едно зарче. На теория има $11^{16} \cdot 6^{16} \approx 1.3 \cdot 10^{29}$ възможни композиции. Много от тях ще бъдат свързани, докато има определена сума от музикален материал, но нито един от тях няма да е съвсем същия. Този брой е толкова голям - 13 следвано от 28 нули, че ако толкова много песъчинки в пясъка и те бъдат подредени в един файл (с презумцията че всяка една трябва да е с дължина един милиметър), те ще обхванат разстояние от приблизително 13 милиарда светлинни години, което е приблизително 130 000 пъти диаметъра на Млечния път. [68]

Този процес включва случайността като важна част от алеаторните процедури използвани от Джон Кейдж два века по-късно, с разлика в тоналността: в играта на Моцарт, последованието от тактове в генерираните менует и трио е трябвало да се съобрази със стилистичните правила на времето. Това е било възможно защото всеки

²⁵ Loy, *Composing with Computers*, 303

²⁶ E.Smith, liner notes to W.A.Mozart, *A Musical dice game* (Phillips Complete Mozart Edition-Rarities and Surprises, 422- 542, 1991)

такт в таблицата е бил избран от поредица от възможни заместващи тактове, аранжирани по такъв начин, че всичките последования да установят и създадат хармонични и стилистични канони.

1.2.2.3 XX ВЕК

През 20-ти век авторите Лорейн и Джоунс представят обяснения на математиката, която се използва в музицирането. До средата на 80-те години, повечето опити за това се фокусират върху тембрите, отколкото върху структурата на композираната музика. Това е разбираемо, впредвид състоянието на изкуството – изчислителните лимити на компютрите и относителната липса на богати звукови източници. Дадените лимити на по-ранните компютри са основателна причина, че повечето композиционни алгоритми са центрирани около нотите, например, веригата на Марков дава вероятности за следващата нота, повече отколкото по-високо структурираните нива, като фразировката или тематичния материал. Нотно-базираните композиции са по-малко изчислителни, но с тяхната леснота идва и организационната размяна.

В днешно време вече има много примери за сложни композиционни алгоритми, които създават хубава музика, повечето от които са разположени около база данни с описания на стилове или правила. От друга страна, компютърът все още често се използва като бърз процесор за произволни процеси. Две известни генериращи програми *M* и *Jam Factory* използват веригите на Марков, за да създават линии, които статистически са близки до това, което потребителят въвежда като данни. Друга тенденция е да се използва компютъра като акомпанимент, който слуша какво се просвирва и отговаря подходящо в реално време. В този случай данните, които въвежда човекът се използват да генерират правила, върху които машината ще базира своя изходен резултат. Това се наблюдава в програми, като *Cypher* и *IBL-Smart*.

Колкото по-мощни стават компютрите, толкова стават по-способни да извършват все по-сложни задачи. Съответно с времето композиторите започват да използват компютрите в по-тесен модел към творческия процес. Това се получава не само в музикалното композиране, но и в цялата област на творчество. Това е неизбежна тенденция, но е също така жизненоважна за утвърждаването на алгоритмичното композиране и за цялото изкуство, генерирано чрез компютър.

За родоначалници на алгоритмичната музика са приети няколко композитора, измежду които са Булез, Ксенакис, Лански, Хилър. [10] Много от тях пишат собствени програми, които използват към конкретна композиция. Ксенакис създава програмата SMP (*Stochastic Music Program*), която се използва и от други музиканти. Лански създава алтернативна поп-музика и реализира алгоритъм за трансформация на английска и китайска реч. В програмите създадени от Хилър и на други композитори на алгоритмична музика се използват основно два подхода, основаващи се на прилагането на детерминирани или стохастични (вероятностни) процедури.

Детерминирани процедури²⁷ генерират музикални събития (например, ноти), изпълнявайки определени композиционни задачи, несвързани със случаен избор. Променливите в такава процедура се наричат изходни данни. Тези данни може да бъдат набор от звукови височини, музикални фрази, правила или ограничения, които трябва да удовлетворяват процедурата. Стохастичните процедури²⁸ въвеждат случаен избор в процеса на приемане на решения [10]. Те генерират музикални събития по вероятностни таблици, установяващи вероятността на събитията. Често е невъзможно по слух да се установи дали даден музикален фрагмент е резултат от стохастичен или детерминиран процес. Следователно, изборът на алгоритъм е въпрос на композиторско виждане. В една система може да има различни типове алгоритми, прилагани към различни измерения (параметри) на композиционния процес.

От гледна точка на творчеството, при изцяло автоматизирани програми за композиция действията на композитора се свеждат до въвеждане на неголям брой изходни данни преди да се стартира програмата, които установяват стратегията му. Един от начините да се избегне зададената стратегия е възможността за въздействие върху логиката на програмата. В този случай композиторът се превръща в програмист и поема пълната отговорност за крайния резултат. Друг начин е възможността за подбор на резултатите от действията на програмата. Най-сериозно ограничение, което се наблюдава в ранните програми за автоматизирано композиране, е пакетния режим на работа [17]. Композиторът подготвя и въвежда изходните данни, стартира програмата, изчаква края на работата ѝ, приема или отхвърля резултата. В пакетния режим минималната единица на композицията е цялата партитура и затова не може да се поправят отделни грешки. Принципът на интерактивност позволява да се възвърне изгубената гъвкавост в работата на системата, като се получава достъп до различни

²⁷ Виж речника

²⁸ Виж речника

слоеве на композиционната структура и до самия процес на композиране: на микрониво може да се влияе върху отделни параметри, ноти, отделни фрази, партии и процедури, а на макрониво може да се влияе върху цялата композиционна стратегия като цяло.

1.2.2.4 СИСТЕМАТА НА ШИЛИНГЕР ЗА МУЗИКАЛНО КОМПОЗИРАНЕ

През 1920 и 1930 г. Джоузеф Шилингер, украински композитор и музикален теоретик, разработва система за музикално композиране, базирана на научни принципи. [66] В неговия главен труд *“The Mathematical Basis of the Arts”*, той се застъпва за формалната теория на естетичното творение.[67] За съжаление, неговите усилия се оказват напразни. Човешката креативност заобикаля всеки вид формалност. Това се оказва труден за решаване проблем и някои автори застават срещу изчислителния модел на креативността, базирайки се на менталните процеси, при креативността и интуицията са неизчислими и не биха могли да бъдат пресметнати.

Системата на Шилингер покрива всички фундаментални аспекти на музикалното композиране, като контрапункт, хармония, ритъм и т.н. Някои от тези аспекти са описани в Приложение 1. Системата е геометрична в своята основа, особено относно концепията за “фазовата взаимовръзка”. Тази концепция обхваща виртуално всеки компонент на системата. Фазовите връзки са в същността на периодичните движения. Неговата методология проектира тези фазови връзки в области на ритъм и структурални пропорции, както и в далеч по-очевидните височинни структури, като тоналности и акорди, контрапункт, акордови прогресии и т.н. Основният принцип, който е заложен в системата на Шилингер е един процес, чиято намеса с обикновени, правилни, симетрични ритмични образци (Фиг.1), произвежда по-сложни, неправилни и несиметрични ритмични образци (Фиг.2). По-долу е даден подобен пример :



Фигура 1. Ритмичен образец [129]

Този ритмичен образец може да бъде представен като обединение на следните два по-прости ритмични образци:



Фигура 2. По-сложен ритмичен образец [129]

Този процес на получаване на сложни ритмични образци от логични операции работи и върху по-прости процеси, наподобяващи начина, по който се синтезират сложните звукови вълни – чрез добавяне на много по-прости, чисти синусоидни вълни.

Работата на Шилингер е критикувана от много автори, аргументирайки критиките си с липсата на каквато и да е научна или математическа основа. Въпреки това, някои негови идеи хвърлят светлина върху нови посоки в композиционните процеси. Той предсказва много от усъвършенстванията в алгоритмичните процеси. Системата му съдържа често срещани препратки към музиката, като “естествена динамика, сила”. Това становище, разбира се, не е нищо ново – то датира от Царлино в ранния Ренесанс, който определя идеята на музиката, като имитация на природата. Употребата на редицата на Фибоначи в системата на Шилингер, с препратки към употребата ѝ в нарастващите образци на растения, черупки, е един от многото примери разпръснати в неговата система на композиране. Това също така е ясно свързано с настоящите разработки в областта на фракталната геометрия, както и в изследванията на хаотичните атрактори²⁹ и нелинейните динамични системи, като модели на естествения феномен. [59] Важна част от дисертацията, която дава ясна представа за употребата на редицата на Фибоначи в музиката, се намира в Приложение 1.

Неговите техники са пренесени в днешно време чрез техните сливания с рекурсивните и хаотични процеси на "фракталите" в музиката – т.е. новите технологии, забележими в увертюрата и "фазовата музика", използвани да възвестяват началото на нова ера в музиката.

Шилингер разработва гигантска система за музикално композиране, както се твърди, основана за първи път в историята на научните закони. Системата се състои от 12 книги, всяка от които обхваща фундаментални аспекти за музикалното композиране, като контрапункт, хармония или ритмична структура. Успехът на системата поне в

²⁹ Виж речника

някои отношения се установява в дългата история на популярната музика и в творчеството на композиторите на филмова музика, които са я използвали – са Джордж Гершуин, Глен Милър, Бени Гудман, Оскар Левънт, Кармин Копола и още множество творци. Всъщност, някои аспекти от системата на Шилингер са толкова широко разпространени, че те основават вид тайно и непризнато течение през цялата музикална култура на Северна Америка (по-специално популярната музика) от 30-те години на 20 век до наши дни. За съжаление, системата, която съществува днес по-скоро е колекция от списъци и обобщения, отколкото ясно тълкуване на включените процедури.

Почти всеки аспект от системата произлиза по отделен начин от фазовите взаимовръзки ("резултатни от интерференцията" в неговата терминология) на прости периодични движения. Той открива начин да проектира тези резултатни в ясни области от ритмиката и структурната пропорция, както и в области от височинната структура (скали и акорди), контрапункт, обертонова прогресия, оркестрация и дори в емоционалните и семантични аспекти на музикалното композиране.

През последните години от живота си, той разширява системата дори в немусикални пространства, за да предостави изображения и дизайн, основани върху същите принципи. Той също така разширява системата, за да достигне пространства на други временни изкуства, включително драмата и киното, развивайки теория на ново кинетично изкуство, основано на съюза между различните изкуства, като музиката и драмата. Оригиналността на Шилингер се крие в направените от него каталози, "върху научни принципи" за всяка възможна пермутация на взаимовръзките между различни форми на изкуството.

Основната идея залегнала в системата на Шилингер е да използва ритмични образци, изведени като резултат от интерференцията на прости периодично повтарящи се примери, както добре познатия размер $\frac{3}{4}$. [66]

Няколко интересни аспекти от тези резултати са загатнати в дискусиите на Шилингер. Първо тяхната дължина винаги е еднаква с продукта на включените елементи. Затова тоталната дължина на резултанта, изразен с ритмичната стойност осмина, ще бъде $\frac{3}{4}$ или 12 осмина ноти.

По подобие, дължината на резултата на $\frac{7}{8}$ ще бъде 56. Тези размери винаги са симетрични; те са според терминологията на Оливие Месиен, не-обърнати, и това е което Месиен нарича "очарованието на невъзможностите".

В много от случаите системата на Шилингер за композиране е вид компютърна музика преди самия компютър. Той предсказва развитието на алгоритмичната композиция, която излиза на преден план много години по-късно. Примери за това са: графични системи за нотирание; техниките на проекция върху музикални параметри; употребата на строги ограничителни методи; теорията за себеподобността; музиката, като имитация на естествената динамика.

Докато някои от тези идеи имат исторически приоритет в системата на Шилингер, като теорията на музиката като имитация на естествената динамика и употребата на редицата на Фибоначи, други имат обстойно и задълбочено изучаване, до научната степен, представена в неговата система .

В подкрепа на тези прости техники, ние дължим на Шилингер теорията на проекцията в много генерален смисъл – решението на проблема за приемане на абстрактните възприятия за валидни. Това той го прави отново и отново фактически във всеки том от системата. За пример, той обсъжда начини, по които всяка от следните математически абстракции може да бъде конкретизирана в мелодия, хармония, ритъм и структура: натуралният обертонов ред; аритметичната прогресия; геометричната прогресия; повдигнатите на степен редици; различни логоритмични редици; прогресивни натрупващи се редици (редицата на Фибоначи); редицата от прости числа; значението на аритметиката; значението на геометрията.

1.2.3 КОМПОЗИТОРИ ИЗПОЛЗВАЩИ ПРИНЦИПА НА НЕСЪЗНАТЕЛНО АЛГОРИТМИЧНО КОМПОЗИРАНЕ

Използвам термина несъзнателно алгоритмично композиране, защото докато изследвах творчеството на композиторите и моделите на техните композиции през призмата на математическия апарат, установих че допреди появата на компютрите, композиторите са имплементирали алгоритмичните методи в творчеството си несъзнателно, следвайки интуитивно естествената връзка между музиката и математиката. Именно поради тази причина в голяма част от настоящето изследване използвам експериментален модел, базиран на познати аксиоми и теореми. В тази връзка в средата на 1970 г. се осъществява едно по-общо математическо проучване на музиката, което е дело на Р. Вос и Дж. Кларк от Калифорнийския университет. Вместо да изучават структурата на музиката, те решават да проучат физичните характеристики на аудио музиката и установяват експериментално алгоритмичната закономерност [53].

По това време американският математик и метеоролог Лоренц вече е създал своята Теория на хаоса, а американският математик Б. Манделброт - своята Фрактална геометрия.[48] Тези нови теории от края на миналия век дават нов тласък в изучаването на музиката от гледна точка на откритията в математиката. Композиторът Литъл изучава алгоритмичния състав на музиката с помощта на теорията на хаоса. Много от неговите композиции са основани на математически модели, които показват хаотичното поведение на музиката, чрез използваното логистично уравнение като вдъхновение.

1.2.3.1 ЕДГАРД ВАРЕЗЕ (*EDGARD VARESE*)

Друг композитор, който играе съществена роля в затвърждаването на лицето на музиката, като изкуство-наука, е Едгард Варезе. Той поддържа древната идея че мястото на музиката е в компанията на математиката и геометрията. Но той също така е наясно, че музиката не е само наука, а че участва и в науката и в изкуството. Той настоява че в основата на креативността стои експеримента, със съзнанието, че необятните възможности на креативността се дължат на новите технологии, които се предлагат в услуга на музиката. [72] “Свободата от установените правила, парализираните темперирани системи; възможността за получаване на какъвто и да е брой кръгове или подразделения на октавата, формирането на която и да е желана тоналност; неподозиран обхват в ниските и високите регистри; нови хармонични съчетания, получени от употребата на суб-хармонични комбинации – вече е възможно ...”³⁰

Едгард Варезе има огромно влияние върху поколението композитори след него, прокарвайки нов път в музикалното композиране³¹. Щокхаузен, Лигети, Булез, Ксенакис (*Stockhausen, Ligeti, Boulez, Xenakis*) и много други нямаше да разцъфнат така пълноценно без Варезе като техен предшественик. (*Приложение 3, Ligeti – Artikulation; Stockhausen Studie II (Elektronische Musik)*)

Основата му в математиката и науката го подготвя за кариерата му като инженер и му помага в неговото търсене за нова естетика на звука. Въпреки че Варезе се е интересувал повече от новия свят на звука, отворен от електронната музика, отколкото

³⁰ E. Varese, *The Liberation of Sound*, In *Source Readings in Music History*, (New York: Norton, 1998), 1343

³¹ Както казва Варезе : “Светът се променя и ние се променяме с него. Колкото повече позволяваме на нашето съзнание романтичния разкош да се превръща в съкровище от спомени, толкова по-неспособни ще сме да срещнем бъдещето и да определим новите ценности, които могат да бъдат създадени в него”.

от приложението на специфични математически процедури в композирането, неговото влияние и идеи обогатяват основата за композиторите, желаещи да експериментират с тези нови техники. [34] (*Приложение 3, Edgar Varèse Poème électronique*)

1.2.3.2 ЯНИС КСЕНАКИС (IANNIS XENAKIS)

Първият композитор, който приема чистия математически подход в музикалното композиране, не просто като инструмент, а като философия на композиране – е Янис Ксенакис. Той е един от пионерите на модерното компютърно-подпомогнато композиране.³² “Линейната полифония се самоунищожава от прекалената си сложност; това което чува един слушател е в реалността, а не маса от ноти в различни регистри.³³ Огромната сложност пречи на аудиторията да следва движението на линиите и има като макроскопичен ефект непредвидено разпиляване по цялото протежение на звуковия спектър. [78]

Ксенакис признава фундаменталното противоречие между линейната полифония и това, което аудиторията наистина чува. Той твърди, че това противоречие постепенно ще изчезне ако свободата на звука стане абсолютна, т.е. когато линейните комбинации на звуците и техния полифоничен строеж няма да продължи да функционира като такъв. “В такъв случай макроскопичния ефект може да бъде контролиран от състоянието на движението на елементите, които се селектират. Резултатът е интродукцията на теорията на вероятността, чиято същност в конкретния случай е комбинаторно смятане.” Ксенакис нарича тази нова парадигма на музиката – “стохастична музика”. Стохастичната музика се отнася до голямо количество звуци и по-скоро до линейна поредица от височини, която се подчинява на тази маса от звуци. Тези звукови събития могат да включват хиляди индивидуални звуци. Линейните редици на тези звуци са неопределени, или по-скоро те не следват никакви видими образци и се разпределят на случаен принцип. Цялостната форма, която създават е добре дефинирана и насочена. [34] Поведението на газа много добре илюстрира тази особеност – газът е съставен от множество газови молекули; мястото и скоростта на всяка индивидуална молекула в газа е непредсказуемо и се основава на случайността, въпреки че цялостната същност на газа не е случайна – той има определена

³² Ксенакис твърди, че “цялостната детерминирана сложност от действието на композиране представлява една слухова и идеологическа безсмислица”

³³ Xenakis, *Formalized Music*, 8

температура и налягане. Индивидуалните височини в тези огромни количества звуци наподобяват индивидуалните молекули в газа: разгледани индивидуално, те очевидно се разпределят на случаен принцип, но разгледани като едно цяло, те създават една добре определена същност. Композиторите могат да контролират как тези маси от звуци да се държат, манипулирайки плътност, скорост на промяна и т.н. Редуцирайки плътността, гъстотата на тези маси от звуци, композиторите могат да постигнат резултати, които наподобяват линейната полифония. Тази нова концепция за музиката е един вид разширение на традиционните полифонични модели. Теорията на вероятността е нужна, за да контролира еволюцията на тези звукови маси. Това включва вероятностните разпределителни функции и непрекъснатите вероятностни функции. С тези математически инструменти, композиторите могат да манипулират и контролират как тези звукови маси се разгръщат и трансформират.

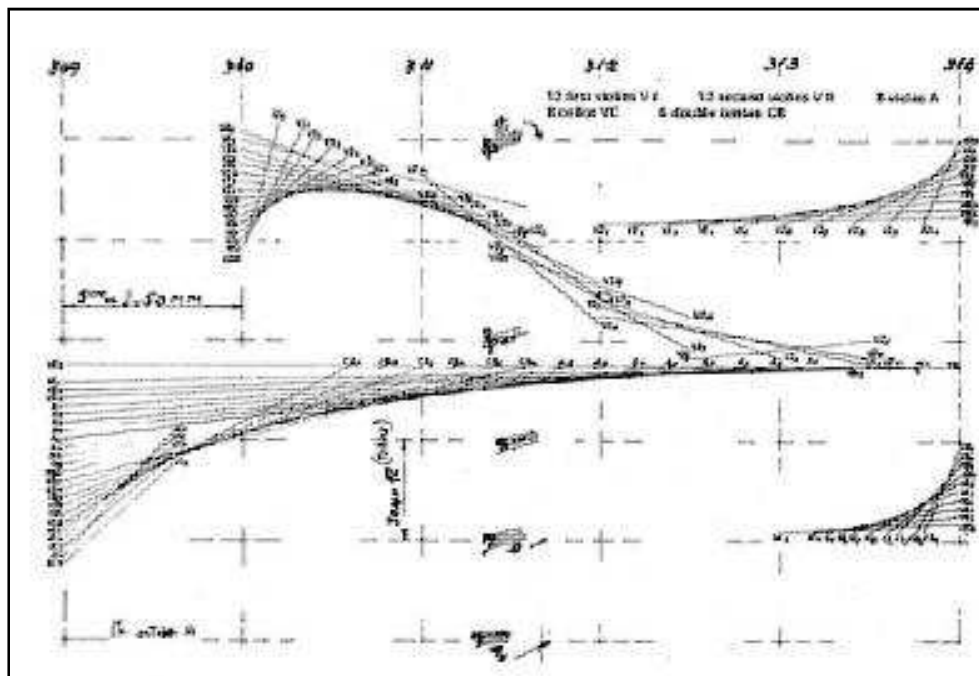
Това се осъществява на макроскопично ниво, върху цялото звуково събитие. Индивидуалните височини, които формират тези звукови събития се аранжират по един неопределен, случаен начин. По принцип, генератори на случайни числа се използват да генерират тоновото ниво на звуковото събитие.

По времето, в което Ксенакис развива стохастичната музика (1950 г.), желанието на композитора да внедри тези техники в музиката е изисквало голямо познание в областта на математиката, понеже всички изчисления са правени ръчно. Компютрите по това време са били лукс и дори да са имали достъп до тях, е трябвало да бъдат програмирани специално за тази задача. Този вид композитори са били хибрид между музикант и математик, с голямо познание и в двете области. В днешно време благодарение на развитието на компютрите, композиторите могат да се сдобият с лесни за употреба програми, които предоставят всички необходими изчисления, спасявайки ги от тези досадни задачи, да пресметят на ръка. Въпреки това все още е необходимо едно дълбоко познание на абстрактния математически модел за пълно експлоатиране на неговите възможности, дори и когато композиторият е освободен от най-трудната част на изчисленията.

Стохастичната музика е критикувана от някои автори заради начина, по който тя ограничава стилистичната обективност на композитора.³⁴ В нея композиторият има контрол над начина, по който се прилага вероятностното разпределение, за да се оформи работата, но няма контрол над финните детайли на звуковата маса. [49] През

³⁴ Loy, *Composing with computers*, 310

1960 г. Макс Матюс заедно с няколко други изследователи, откриват начини да се преодолее това ограничение чрез употребата на детерминирани алгоритми и квантазиране на височината на тоновете. В един от техните експерименти, те създават контрапунктова структура, в която интервалите между гласовете са били квантазирани до най-близките терци, кварта, квинти и сексти. Броят на гласовете и метода за височинно генериране могат да бъдат сменени при желание от композитора. Изследвани са различни стратегии, включително различни нива на височинно квантазиране – от абсолютна честота (неквантазирана) до дискретно квантазиране в темперирани и диатонични строи. Техните методи позволяват на композиторите да имат контрол над тоновото ниво на творбата, продължавайки да поддържат философията на стохастичните процедури.



Фигура 3. Янис Ксенакис прилага теория на вероятностите и математически модели към композирането на музика. Графичната партитура на *Metastasis* (1964) визуално дефинира природата на звука, която трябва да бъде изпълнена и имаща естествена последица в композирането на електронна и компютърна музика [130]

Един забележителен пример в творчеството на Ксенакис е оркестралната творба *Metastasis* (1964), графична партитура на която може да видите на фигура 3, изградена около множество от независими инструментални глисанди.³⁵ Творбата изисква оркестър от 61 музиканта, всеки от които трябва да свири различна част от формата на плъзгащите се ноти. Работейки на компютър IBM 7090 Ксенакис композира много

³⁵ Нотната партитура на творбата на Ксенакис “Metastasis” може да видите в Приложение 5, Iannis Xenakis.pdf

творби, основани на математиката. (*Приложение 3, Iannis Xenakis – Diamorphoses; Iannis Xenakis – Metastasis*)

1.2.3.3 ЛЕДЖАРЪН ХИЛЪР (LEJAREN HILLER)

Той е един от първите композитори, които усвояват един систематичен, алгоритмичен похват на композиране и както се твърди – първият човек, който композира музикална творба с помощта на компютъра. Неговото задълбочено познание в областта на науката и математиката (придобива докторска степен по химия) му осигурява необходимото познание, с което да осъществи своите идеи. Музикалното мислене на Хилър е силно повлияно от теория на информацията.

Теория на информацията е клон на приложната математика и електротехниката, включващи количествено определяне на информацията. Информация теория е разработена от Д. Клод Шанън, за да се намерят основните ограничения за обработка на сигнали, компресиране на информацията и за надеждно съхраняване и съобщаване на данните. От самото начало тази теория е разширена, за да намери приложение и в много други области, включително статистически извод, обработка на естествен език, криптографията като цяло, мрежи, различни комуникационни мрежи, както и в невробиологията, в развитието и функциите на молекулярните кодове, модел за подбор в областта на екологията, топлинна физика, квантови компютри, за откриване на плагиатство и други форми на анализ на данните. Като цяло теорията на информацията се отнася до асимилация, уподобяване, количествено определяне и оптимизиране на ефикасността на информационния трансфер. [29]

Тя е важна не само за модерните комуникационни технологии, но също така и за разбирането на езика и трансмисията на информация от други процеси. В приложението на теорията на информацията в музиката, Хилър внася флукси (непрекъснато изменение) в общата сума от информация, която се предава в една творба към слушателя, за да достигне до съществената драматична природа на музиката. Тази предпоставка основно повлиява на начина, по който композира Хилър. През 1957 г. Хилър написва *The Illiac Suite*, известна също така като *The Illiac String Quartet* или по-просто като *String Quartet №4*. Тази творба, генерирана от ILLIAC компютър в университета на Илинойс, с помощта на Леонард Исааксън и Робърт Бейкър, е първия пример за композиция, създадена с дигитален компютър.

ILLIAC е серия от суперкомпютри създадени в най-различни места, някои в университета на Илинойс в Урбана-Шампейн. Общо пет компютъра се създават от тази серия между 1951 и 1974 година. Някои по-съвременни проекти продължават да използват названието.

През 1963 г. заедно с Робърт Бейкър, Хилър създава компютърна програма на език за композиране, наречена MUSICOMP, която използва за композирането на *Computer Cantata*. Няколко от командите, които са на разположение в тази програма са структурирани именно с идеята на теорията на информацията. (*Приложение 3, Hiller Computer Cantata*)

Тази програма е използвана и за създаването на други творби, като *Algorithms I, Algorithms II и Algorithms III*. (*Приложение 3, Lejaren Hiller Algorithm I (Versioni I and IV)*)

Цялостното творчество на Хилър разкрива един силно еkleктичен композитор, участващ във всички музикални събития на своето време, от популярния стил джаз до трудния за разбиране сериализъм. (*Приложение 3, Lejaren Hiller Machine Music*)

1.2.4 СЪВРЕМЕННИ ТЕХНИКИ

С появата на теорията на хаоса (*Chaos theory*)³⁶ и с развитието на други научни дисциплини, като клетъчна автоматизация (*Cellular automata*)³⁷ и фрактална геометрия и благодарение развитието на изчислителната техника, започват да се установяват нови връзки между музиката и математиката.

1.2.4.1 ТЕОРИЯ НА ХАОСА

Теорията на хаоса, въведена от френския математик Анри Поанкаре (*H. Poincare*), в началото на 20-ти век достига своя връх през 1980 г., като описва непредсказуемото поведение на системи, които са повлияни от едно общо положение или поредица от условия. [27]

До неотдавна, учените вярваха, че случайните влияния карат системите да се държат непредсказуемо. Те вярваха, че ако премахнат случайните влияния ще могат да предвидят поведението. Сега те вече знаят, че много системи могат да проявят

³⁶ Виж речника

³⁷ Виж речника

непредсказуемост в дълъг интервал от време, без да бъдат повлияни от случайни фактори. Подобни системи са хаотични. Хаотичните системи са непредсказуеми, защото са чувствителни към своите първоначални условия, като позиция и скорост. Две идентични хаотични системи, настроени за движение с малки различия в първоначалните условия могат да възпроизведат много различни движения. Тези системи описват толкова сложни и непредсказуеми поведения, че са достойни за описанието «случайност». Поради сложността и броя изчисления, включени в изследванията на хаотичните системи и благодарение на възхода на изчислителната мощ през последните 20 години, теорията на хаоса достига своя истински връх.

Математическите уравнения, които моделират поведението на хаотичните системи, са парадоксално извънредно прости. Те се наричат нелинейни уравнения. Нелинейните уравнения имат безкрайни решения. Характерна черта на тези уравнения е итерацията: решенията се връщат обратно в променливите на уравнението рекурсивно. Итерацията позволява на тези относително прости математически системи да моделират хаотичното поведение на много естествени процеси. Начинът по който математическите системи се употребяват в музикалното композиране е чрез кодово-графично преобразуване на цифровия резултат на уравнението към музикални параметри.

Теорията на хаоса очарова композиторите и много от тях, като Чарлс Уоринън, Гари Лий Нелсън, Дейвид Кларк Литъл, я адаптират към творчеството си като инструмент за композиране.

1.2.4.2 КЛЕТЪЧНА АВТОМАТИЗАЦИЯ (*CELLULAR AUTOMATA*)

Друга дисциплина, за която неотдавна бе доказано, че дава полезни резултати в музикалното композиране е клетъчната автоматизация (*cellular automata*). Клетъчната автоматизация е официално представена в средата на 60-те години от Джон фон Нойман, като модел за компютърно-симулирано биологическо себепроизводство. Този модел се състои от двуизмерна мрежа от клетки, в която всяка клетка позволява число от различни състояния. Клетките сменят своето състояние в мрежата, съобразно дефинирана редица от правила. Стартирайки с първоначалната конфигурация на клетките – основно наричани генерация 0, в която всяка клетка може да има всяко от позволените състояния, а редицата от правила се прилага върху всички клетки от мрежата, възпроизвеждайки нова конфигурация от правила (генерация 1), в която

отново се прилагат редицата от правила и т.н. В зависимост от броя на позволени състояние на клетки и редицата правила, могат да бъдат генерирани различни клетъчни автоматизации, като някои от тях показват голямо богатство, докато други замират много бързо, а трети показват възможности за себепродуциране: при определени условия, някои конфигурации на клетки са способни да се регенерират генерация след генерация. [77] Тези характеристики са много интересни от композиционна гледна точка. Традиционно, композиторите използват разпространените образци интуитивно: редиците са добър пример за разпространение. Клетъчната автоматизация при алгоритмичните процедури, приложени към музиката, позволяват разпространението, размножаването на образа да бъде формализирано на по-високо ниво. [74]

Клетъчната автоматизация е доказано, че е обширна област за алгоритмичното композиране. Композиторият Едуардо Миранда [56], разработва компютърната програма за РС, CAMUS, която преобразува двуизмерна клетъчна автоматизация в музикални параметри.³⁸ Неговата творба *Entre l’Absurde et le Mystere* е композирана с тази програма. [55]

1.2.4.3 Л-СИСТЕМИ (L-SYSTEMS)

Друга област от математиката, която неотдавна бе внедрена в алгоритмичното композиране е Л-Системата (*L-System*). Л-Системите първоначално са предложени от Аристид Линденмайер през 1968 г., като основа за аксиоматична теория на развитието.³⁹ Впоследствие те се използват за моделиране на живи организми. Основно, Л-Системите се състоят от редица от заместващи правила, рекурсивно приложени към първоначалния низ от символи (обикновено по-сложни), като структурни елементи на организма. [64] Заместващите правила определят как всеки символ в текущата генерация ще бъде заместен. [47] Поради богатата структура на генераните Л-Системите, те могат да бъдат интересен източник за алгоритмично композиране. Някои композитори вече успешно прилагат Л-Системите в композициите си, като Гари Лий Нилсън в неговата *Summer Song for solo flute*.

³⁸ E.R.Miranda, “Cellular Automata Music: An Interdisciplinary Project”, *Interface* 22 (1993): 3-21; E.R.Miranda, “Music Composition Using Cellular Automata”, *Languages of Design* 2 (1994), 105-117

³⁹ Aristid Lindenmayer, “Mathematical Models for Cellular Interactions in Development, Parts I-II, “*Journal of Theoretical Biology* 18 (1968), 280-315

1.2.4.4 ГЕНЕТИЧНИ АЛГОРИТМИ (*GENETIC ALGORITHMS*)

Генетични алгоритми (*Genetic Algorithms*) са едни от най-късните математически техники, приложени към алгоритмичното композиране.⁴⁰ Един генетичен алгоритъм е процедура, която търси потенциално обширно решение за оптимално решение на даден проблем. [39] Решенията са кодирани като низове върху ограничена азбука. Битовите от най-подходящите решения се използват да генерират нови низове решения. Всяка стъпка, генерация на един алгоритъм произвежда нова популация от възможни решения, базирани на популацията от предишна генерация. [28]

Когато генетичните алгоритми се приложат към музика, се следват тези процедури. Редиците (или решенията) кореспондират към музикални същности, като музикални мотиви или дори цели фрази. Първоначалната популация на решенията (генерация 0) може да бъде дадена от композитора или да бъде генерирана от самия компютър чрез стохастик средства. Необходимо е да бъдат направени правилни тестове от композитора. Те определят кои решения (музикални същности) ще оцелеят и кои ще умрат. Когато се случи една оптимална редица от решения – един процес, чиято дължина зависи от първоначалната популация, пространството за решение и удачен тест – тогава програмата спира. Генетичните алгоритми инкорпорират техники директно от естествената генетика:

- Естествена селекция: редици с подходящи стойности оцеляват от една генерация до друга, с висока правдоподобност. Редици със слаби стойности не успяват.
- Репродукция: две редици избрани чрез естествена селекция се свързват (чрез *crossover* – пресичане, кръстосване, който подбира мястото в редицата на случаен принцип и обменя информацията между двете редици), за да възпроизведат нови редове. Музикално, това е аналогично на конструирането на нова фраза или мотив от няколко други.
- Мутация: редиците могат да претърпят спонтанни промени (с малка вероятност), за да възпроизведат нови редици (нови музикални същности) в различна част.

⁴⁰ D.Goldberg and A.Horner, "Genetic Algorithms and Computer-Assisted Music Composition" in Proceedings of the 1991 International Computer Music Conference (San Francisco: International Computer Music Association, 1991), 479-482

Тези операции изискват употребата на вероятностни функции и заради това имат стохастични черти в природата си.

1.2.4.5 ДРУГИ ТЕХНИКИ

Други математически модели, които са адаптирани за алгоритмично композиране през последните години включват шум (специално т.нар. $1/f$ noise или розов шум), числова теория (числови редици, редицата Морз-Тю (*Morse-Thue*) и фракталите [18]. Предишните успехи на процесите до голяма степен определят най-важните методи, които се използват в алгоритмичното композиране в днешни дни. Колкото повече нараства изчислителната мощ и се развиват нови научни дисциплини, толкова повече съществуващите алгоритмични процедури тепърва ще бъдат изследвани и нови такива ще бъдат инкорпорирани. Тези техники са един истински инструмент, чийто възможности са винаги на разположение на композитора. Материалът, който генерират може да бъде разгледан като музикално “основно съдържание”, което по-късно може да бъде модифицирано, трансформирано и инкорпорирано съобразно желанията и естетичните цели на композитора. Алгоритмичните методи в музиката трябва да бъдат използвани основно като източник на вдъхновение, а не като едностъпково музикално решение.

1.3 ФРАКТАЛНА МУЗИКА

Основен обект на настоящето изследване е фракталната музика. Появилото се през последните години в науката ново течение и област за изследвания, свързано с хаоса, фракталите и себеподобността, даде нови отправни точки за проучвания, които привличат общественото внимание не само с прекрасните картини, които произлизат от тях, но и с музиката, която създават. Едва ли има някой, който да не познава картините на Манделброт или да не е чувал за имажинерните числа в математиката.

Фракталната музика може да бъде разглеждана по два начина: като частен случай на алгоритмичната музика, използваща същите алгоритмични уравнения, които генерират фрактални образи или като съвсем отделен дял, касаещ музиката, разбираана като фрактален модел.

Терминът фрактална музика се отнася за музиката, композирана изцяло или частично, използвайки същите типове от повтарящи се или рекурсивни процеси, които

се използват за създаване на фрактални образи. Често става така, че музиката се композира директно от фрактални изображения. Компютрите композират фрактална музика използвайки винаги формални алгоритми - софтуерни програми, за да създават музика.

В еволюцията си алгоритмичната музика достига до фракталната музика, в основата на която се намират фракталните алгоритми. Фракталната музика е резултат от итерационни процеси, при които един и същ алгоритъм се прилага многократно към предходни резултати.

Създаването на музика от фрактали е пример за композиране на алгоритмична музика. Една музикална идея трябва да бъде преведена и трансформирана в съдържание, като разликата между традиционните принципи на композиране и тези, които използват за композиране определен софтуер, е в самата идея, която цели да внуши композитора.

Любителите на класическата музика ще останат изненадани от един факт, който трудно се възприема от музикантите и той е че класическата музика, както и днешните популярни стилове и музикални форми са фрактални в същността си. А това е свързано и се обяснява най-вече със свойството на фрактала да се самоповтаря и себеизразява. Самите тонове имат тази способност, неопределяна досега с точно тези термини. Тоновите, които се композират и прозвучават в конкретна последователност и принцип на мелодична линия, която следват и прозвучават в отделни тактове или части от произведението, всъщност следват един основен принцип – себеподобността и детайлизираната вариация.

Техниката за създаване на фрактали е толкова многостранна, че може да се използва за създаване на музика във всеки един стил и има невероятен потенциал за добавяне на нови, вълнуващи елементи към музиката, която композирате .

Уравненията, които генерират фрактали, по принцип са нелинейни. Нелинейните уравнения включват итерация, която представлява нееднократно връщане решението на уравнението в самото уравнение. Интересен е факта, че нещо създадено от семпла математическа процедура, може да бъде толкова естетически приятно

Експериментите, чрез които учените се опитват да покажат красотата на фракталното изкуство, разширено и в електроакустичната музика, понякога дават интересни резултати, но често звучат твърде странно.

Смятам, че сложността произлиза от факта, че теорията за хаоса обикновено работи с реални числа, докато музиката, която използваме се базира на дискретните честоти – скали на честотите.

1.3.1 СЪЩНОСТ НА ФРАКТАЛНАТА МУЗИКА

Всеизвестен е факта, че откакто съществува музиката, оттогава датират и опитите тя да бъде създавана по различни начини и чрез различни композиционни методи. Различният подход днес към музиката се дължи на появата на компютрите. Пред музиката се отвориха хиляди нови методи и средства, които да я създават – основният от които се оказа компютъра.

Много критици твърдят, че компютрите не могат да създават музика със същото качество, както големите композитори, докато други смятат, че това не е най-важния аспект при създаването на музика. Важното е тя да бъде създавана.

Когато слушаме алгоритмична музика - тоновете, генерирани от тези алгоритми, ще бъдат разпознати като приятна мелодия, докато друг ще ги възприеме като хаотична музика. Ако трябва да се класифицира фракталната музика, като някакъв определен стил от съвременната музика, то възможно название може да бъде “новата” или “модерната” музика на 21 век, подобно на музиката на Шьонберг, в чиято невероятна дисонантност и неопределено мелодическо движение, стои забележима пропорция и принцип на хаотичността.

Компютрите и технологиите, които се използват за генериране на алгоритмична музика могат да създават хиляди такива хаотични поредици от тонове – на това ниво в алгоритмична музика липсат компоненти, като динамика, темпо, трайност, тембър. Това вече може да бъде добавено към даден скрипт или да се използва конкретно приложение за създаване на музика.

По-съвременното откриване на фракталите и особено, тяхната връзка с музиката, отвори нова врата за композиторите, зад която се откриха необхватни възможности за творчество и композиране.

1.3.2 ИЗЧЕРТАВАНЕ НА МУЗИКА (КОДОВО-ГРАФИЧНО ПРЕОБРАЗУВАНЕ)

Тайната, която се крие зад фракталната музика, включва нещо, което математиците наричат кодово-графично преобразуване (*mapping*). С оглед на факта, че този термин няма български еквивалент, въвеждаме понятието кодово-графично преобразуване, което изразява същността на процеса. Кодово-графичното преобразуване представлява създаването на директни връзки между числовите уравнения на изходните данни и точни параметри, които могат да използват тези изходни данни. [36]

Кодово-графичното преобразуване се състои от установяване на връзка между данните на математическия модел (обикновено цифров) и една поредица от музикални структури (като тоналност, ритмични стойности, динамика). По този начин изчертаването създава тясна връзка между един нов математически процес (математическия модел) и установените музикални структури. Избора на тези музикални структури за процеса на изчертаване се определят от композитора. Най-простите редици, които демонстрират процеса на изчертаване на математически модели са: гами (обикновено са хроматични), прости ритмични стойности и динамика. Тези редици се използват поради факта, че тяхната простота позволява на структурите, постигнати от математическите модели да могат да се разпознават по-лесно. Вместо гами и прости ритмични форми, могат да се използват постоянна честота, ритмика и динамика, което позволява микротоналност и неограничена сложност на ритмиката, без това да афектира върху вътрешната структура постигната от математическия модел. Важно е да се подчертае, че математическите модели предлагат на композитора нови подходи за вариантност и музикално усъвършенстване.

1.3.3 ДВОИЧНИЯТ КОД НА НОТНОТО ИЗРАЗЯВАНЕ

Класическите композитори никога не са мислели при композирането си за преобразуването на техните ноти, ритъм и музикални експресии – с термини от математиката. Те са били по-склонни да отнесът музиката към господстващата литература и поетическите аналогии на тогавашните времена. Въпреки това, може да се забележи, че техните композиционни сборници съдържат математически паралели, дори когато се е ограничавало влиянието на алгебрата.

Разглеждаме едно елементарно кодово-графично преобразуване между математическите уравнения и музикалните ноти: срещу всяка нота от музикалната хроматична гама съответства едно число.

c 1	c# 1	d 1	d# 1	e 1	f 1	f# 1	g 1	g# 1	a 1	a# 1	etc...
1	2	3	4	5	6	7	8	9	10	11	

Искам да уточня, че това е просто кодово-графично преобразуване на MIDI (*Musical Instrument Digital Interface*) числа към тези полутонове. Тези числа могат да представят актуални ноти, изсвирени на синтезатор; изходните данни от магнитен носител или просто теоретична насока.[54]

Математиците могат да създадат серия от музикални ноти, подредени на принципите на музикалната гама – използвайки константна функция за квантазиране на дискретния сигнал. Това събитие е константа, защото е едно и също между началото и края на събитието. Нотите са константни, освен за момента между различните ноти. Квантазирането означава в този случай, че нотите ще попаднат на точно фиксирани нотни стойности, на абсолютно точни и неизменни точки във времето.

Нека да вземем като пример при една графика, където хоризонталната ос представя времето, а вертикалната представя нотата или броя музикални събития. При това числово/музикално кодово-графично преобразуване, обикновени мелодии (или колекции от музикални събития) се подобни на асиметрични функции; асиметрични, защото събитията не се случват от лявата страна на вертикалната ос в тази координатна система. Те обикновено ще преминат през пресечната точка на времевата ос $= 0$.

1.3.4 ОТ MIDI КЪМ 12 TET (*TONE EQUAL TEMPERAMENT*)

12 TET преведено на музикален език това съкращение означава, че дадено музикално произведение използва т.нар. равномерна температура или равномерно темперирани строй. Фигурата по-долу (Фиг.4) показва MIDI спецификациите, които превеждат нотите по свой критерий и дефинират тонът До или "Middle C" от 1 октава с числото 60, като всички останали тонове се определят спрямо този тон. В тази фигура не е спазена класическата номерация на октавовите групи и затова **Middle C = C4**, докато правилното класическо изписване на този тон щеше да е **Middle C = c1**. По-долу

са дадени няколко популярни мелодии, за да се покаже как изглежда превода им от музикален език към Midi.

Октава	MIDI числа											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Фигура 4. Midi числа [131]

1. MARY HAD A LITTLE LAMB – популярна песен

1 4B1000 4A1000 4G1000 4A1000 4B1000 4B1000 4B2000 4A1000 4A1000 4A2000
4G1000 5D1000 5D2000 5C1000 5D1000 4B2000

2. Mozart – лайт мотив от “Малка нощна музика”

25 4G750 4D250 4G750 4D250 4G250 4D250 4G250 4B250 5D1000 5C750 4A250 5C750
4A250 5C250 4A250 4F#250 4A250 4D2000

3. Beethoven – лайт мотив от Симфония №5

37 4G500 4G500 4G500 4D#2000 4F500 4F500 4F500 4D2000

Може също така да се преобразуват числови стойности към други скали или групи от ноти, за да се постигнат различни резултати. Характеристиката на тези фрактални мелодии е малко объркваща. Ако се анализират в детайли – ще се открият всички типове на сходствата и взаимовръзките между настройките на тоновене и понеже има безкраен обхват от числа, между които да се избере, може да имате неизчерпаем източник от мелодии.

Фракталната музика е резултат от един повтарящ се процес, при който един алгоритъм се прилага множество пъти, докато се достигнат предишните изходни данни

на самия процес. В една по-широка перспектива всички музикални форми, както на микро, така и на макро ниво, могат да бъдат моделирани с този процес. Фракталите осигуряват изключително интересни музикални резултати и тази област на изследвания е основно застъпена като обект на анализ в настоящата дисертация.

Теорията за фракталната музика обсъжда възможностите за потенциална употреба на рекурсия, итерация и комплексна математика, като вид разширяване на традиционната музика и на композиционните практики.

ПРИНОСИ

В първа глава от дисертацията е направено историческо изследване върху развитието на компютърната, алгоритмичната и фракталната музика. Дадени са подробни определения за компютърна, алгоритмична и фрактална музика. Направен е сериозен исторически обзор на развитието на компютърната и алгоритмичната музика, както и сравнителен анализ върху спецификата на компютърната, електронната и електроакустичната музика.

Приведени са сравнителни теми за това как музиката бива превеждана на различни езици – използвайки музикалните закони за структурно съдържание или използвайки протокола MIDI за комуникация между нотите и компютъра.

Алгоритмичната музика обхваща една нова посока в музикалното композиране. Тази посока винаги е присъствала в музиката, по един неназован досега начин. Днес, с помощта на компютърните технологии, можем да продължим да изследваме тези начини и път на композиционно мислене.

В тази глава въвеждам термина кодово-графично преобразуване, поради факта, че термина мапинг (mapping)⁴¹ няма български еквивалент.

⁴¹ Виж речника

ГЛАВА 2. НОВИ ПОДХОДИ ЗА СЪЗДАВАНЕ НА КОМПЮТЪРНА МУЗИКА

Понятието компютърна музика по смисъла на това изследване включва цялата музика, чиято обработка се извършва посредством изчислителни действия. Що се отнася до многообразието на музиката, тя може да бъде обработена, изследвана и създадена чрез компютър, като отделните стилове и жанрови особености имат своите характеристики и параметри. Съществуват в световната литература много прецизирани определяния на това понятие, което диференцира жанровото и стилово разнообразие, но към днешна дата не е възможно да се даде точно определение на понятието компютърна музика, съпоставяйки го с еквивалентното понятие, създадено през 60-те години.⁴²

«Компютърната музика се различава значително от електронната музика. Понякога може и да звучи като нея, но терминът електронна музика обикновено се използва за звукове, произведени чрез електронни средства, с магнетофон, аналогови синтезатори и електронни музикални инструменти. Компютърната музика се появява сравнително по-късно. Едва през последните няколко години тя напусна изследователските студиа и институти, за да се превърне в част от цялостната структура на музикалната техника.» [17]

Компютърът има четири роли, които включват композиране, компютърен синтез, компютърен контрол над функциите, свързани с аудио обработката и изпълнение и на четвърто място - компютърно семплиране⁴³ на аудио сигнала. [80]

Всичко което ни заобикаля в съвременността се създава чрез компютърните технологии. Всичко което чуваме, виждаме и дори усещаме - използва компютрите, за да стигне до нас. Всичко което наричаме изразни средства на изкуството се създава чрез компютърни технологии. Няма композитор, който вече да не борави с компютър и да не използва компютърните технологии, с чиято помощ да създава своите творби. Няма художник, който да не се занимава и с графичен дизайн, нито артист, който да не използва мултимедията, за да привлече зрителското внимание. Изкуството днес е

⁴² Компютърната музика е течение в електронната музика, основано на използването на компютърни технологии. Компютърната музика е музика, която не може да бъде създадена без употребата на компютри.

⁴³ Семплиране – високоскоростен цифров запис...представляващ цифрова мостра на звука, който се записва. С прилагане на транспониране семплиране се явява най-модерната технология за обработка на звук.../3, с.245/

подчинено на компютрите. Нещо повече - изкуството вече се нарича дигитално. Можем само с носталгия да си припомним от тоновете изписани истории на изкуството как се е създавало то през вековете. Днес изкуството е неизменно свързано с мощен процесор, голяма памет, добра видео платка и външна звукова карта. Иначе не би било чуто, видяно или разбрано. Въпреки кратката си история, компютърната и алгоритмична музика поставиха началото на една нова ера в правенето и разбирането на музиката. Развитието на компютърните технологии наложи тяхното бързо развитие и адаптиране към новата сетивност на поколението, което започна да расте с компютър в ръка, а не с играчка. Това разбира се даде предпоставка за развитието на нови подходи и методи за създаването на новата компютърна и алгоритмична музика. Колкото повече се развива компютърния свят, толкова повече методи и средства се създават и в света на музиката.

Днес всеки, който е изкушен от необятния свят на музиката и който притежава компютърни умения и качествен софтуерен продукт, може да претендира, че е автор на музика. Този автор не е посветил безброй часове от детството и юношеството си на любим инструмент или музикална теория. Новите композитори на 21 век притежават всички тези познания и опит в един компютър и добра музикална програма.

Всичко това знаем, че е в резултат на еволюцията и иновациите в съвременните технологии.

Вероятно някой ще попита не е ли именно в това силата и магията на музиката от древните Гърция, Индия и Китай, през барока и класиката, през романтизма, импресионизма и експресионизма?! Че е била създавана от гения на един човек, върху един нотен лист. Великите композитори явно са имали в своя компютър заложен чип от гениалност и талант, за да можем днес да правим успешни аналогии между техните произведения и съвременните музикални тенденции, осъществени чрез фрактали и алгоритми.

Всеизвестно е че Бах и Моцарт /по думите на близките му - гениален математик/, Берлиоз и Вагнер, Лист и Верди, са създавали бушуващ да се излее от тях на нотния лист водопад от красота, като са я вкарвали в стройна форма-двуделна, триделна, многопластова, за да зазвучат безбройните обертонове на истинското изкуство. И всяка една тяхна музикална картина е отворила сетивата ни към едно безкрайно развитие и мултиплициране.

Ако за момент си представим, че в ръцете на горепосочените гении имаше и мощен компютър, може би щяхме да видим как света днес слуша притихнал надигация се музикален тайфун.

Тези велики гении са ни дали посоката и когато искаме да творим, неизменно отново и отново се вглеждаме в богатата оркестрова багра и великолепието на инструменталното многообразие, което са ни завещали в милиони нотни листа, пред които винаги ще се благоговее, защото никога няма да бъдат повторени!

2.1 АНАЛИЗ НА НОВИТЕ ПОХВАТИ ЗА СЪЗДАВАНЕ НА КОМПЮТЪРНА МУЗИКА

Интересно е да се запитаме какво кара композитора да мисли по различен начин, когато композира с компютър, или го прави по традиционния начин с нотен лист и молив? Когато композиторите използват компютър да творят, самата машина улеснява тяхната работа, като извършва автоматично всички изчисления и операции. Вярно е, когато говорим за приложението на компютрите в музиката, се оказва, че те са неизменни помощници. Има основно два различни типа софтуер за музикално композиране: софтуер за алгоритмично композиране и софтуер за компютърно-подпомогнато композиране. Докато софтуера за алгоритмично композиране е програмиран да генерира музика с определена автономност, софтуера за компютърно-подпомогнато композиране служи като средство в помощ на композитора, за да организира и подреди своите идеи. Важно е самите композитори да разберат сами за себе си разликата между тези два подхода на композиране. В следваща глава от дисертацията ще обърнем повече внимание на първия тип софтуер, защото въплъщава в себе си музикални системи с генеративна природа. MIDI секвенсерите, софтуерните синтезатори, семплерите и различните компютърно-базирани музикални софтуерни програми се отнасят към втория тип софтуер и не са обект на изследване в тази дисертация.

2.1.1 ДИГИТАЛЕН СИНТЕЗ НА МУЗИКАТА

Развитието на модерната електронна музика и методите за синтеза ѝ вървят паралелно с развитието на компютърните технологии. Основната цел на компютърната музика, откакто е поставено нейното начало, е развитието на музикален програмен

език и метод за директен синтез на цифровите сигнали в звука. Директният синтез на звука се базира на цифрови алгоритми, създадени за тази цел и на дигитални тон-генериращи поредици. Тонът може да бъде възпроизведен директно чрез съответстващ MIDI инструмент или да бъде генериран от софтуерно базиран синтезатор [54]. В компютъра се използват аналогово-дигитален и дигитално-аналогов преобразувател, които преобразуват съответно аналогов звук към дигитален и дигитален към аналогов. DAC се използва да конвертира дигитални цифрови кодове в аналогови електрически вълни, които да прозвучат от даден високоговорител. Електронните музикални инструменти могат да бъдат контролирани чрез MIDI или подходящ компютърно базиран софтуер. Софтуерът за музика се използва, за да моделира височината, тембъра, амплитудата, продължителността и силата на звука, който се възпроизвежда от инструмент, свързан с компютъра.

В дигиталните музикални системи величините, които се използват, за да се представяват честотата, амплитудата, тембъра и продължителността на звука, са дискретни числа. Числата се въвеждат и изчисляват, за да се постигне желане резултат, като нарастване на силата или промени в темпото .

Звукът в реалния свят се формира от продължителни акустични звукови вълни. Една дигитална система конвертира тази аналогова звукова вълна в числови, двоични данни, които да могат да бъдат съхранени, обработени и възпроизведени отново като аналогична звукова вълна в пространството.

Едно двоично число е число, за което всяка индивидуална цифра може да съдържа две стойности: 0 или 1, или On и Off. Компютрите са създадени да интерпретират и манипулират обикновени десетични числа, които се съхраняват като двоични, използвайки 1 и 0. Една двоична бройна система е идеална за използване от компютрите, защото дигиталните електрически вериги съществуват само в две състояния: On или Off. Едно двоично число се създава от която и да е последователност от двоични числа или битове, като всяка цифра от тях може да се представи от 0 или 1. Един образец на битове включва съдържанието на команда или инструкция . За пример, два бита могат да приемат 4 различни конфигурации —00, 01, 10, 11— предоставяйки компактен метод за предаване на различни стойности в двоичен код. Един байт (*byte*) е универсално приета конвенция за създаване на двоични кодове и се състои от 8 бита(*bits*), правейки възможни 256 различни стойности с всеки бит. Компютърните команди, като командите за MIDI, могат да съдържат един или два байта, в зависимост от спецификацията на използвания програмен език. [81]

Компютрите работят на базата на предоставени процедури (алгоритми), които да могат да бъдат организирани като редица от команди, използващи двоичен код. Един програмен език се използва, за да комуникира между командите и компютъра. През годините са измислени много различни програмни езици за създаването на приложения за компютърна музика, но както отбелязах по-горе, първият основен музикален програмен език за компютър е развит от Макс Матюс.

За създаването на звук, чрез средствата на компютъра и за развитието на дигиталните синтезатори – през годините са създадени няколко метода. Те са директен дигитален синтез (*direct digital synthesis*), цялостно семплиране (*complete sampling*), нотно семплиране (*note sampling*) и синтез на звукова вълна (*wavetable synthesis*). Тези методи за синтез на музиката не са предмет на изследване в настоящия дисертационен труд.

2.1.2 МУЗИКАТА НА КОМПЮТЪРНИТЕ ТЕХНОЛОГИИ

Музиката, която започна да се създава чрез компютърните технологии, се превърна в ключова експресия – начинът за себеизразяване на културата от 21 век. Много хора в днешно време имат възможността да създават дигитална музика чрез разпространените компютърни технологии, интернет и софтуерните приложения, които позволяват създаването на музика (включително софтуерът, създаден от академичната общност през последните 30-40 години, който сега също е в публичното пространство). Това усъвършенстване се отразява върху разрастващата се съвременна музикална сцена и достига нови постижения в музикалния материал и музикалната композиция.

През следващите десет години инженерните проучвания в областта на дигиталната музика ще обхванат разгръщащите се музикални форми и нови възможности за музикално изразяване и разпространяване (включително внедряване на нова звукова култура и прилагане на нови музикални форми). Това ще подсили приложимостта на новата музикална култура към новите естетически и културни усъвършенствания, които си пробиват път през последните години – позволявайки на изследователите да откриват решения за съвременните музикални проблеми и приложения. [20]

Много от въпросите относно проучванията в областта на дигиталната музика, които бяха възпрепятствани в миналото поради слаба подготовка при представянето на

аудио приложенията, слаби възможности при съхраняването на дигиталното аудио и интерактивни аудио приложения, които не са в реално време, сега вече са разрешени.

2.1.3 КОНЦЕПЦИЯ ЗА СЪЗДАВАНЕТО НА ДИГИТАЛНА МУЗИКА ЧРЕЗ КОМПЮТЪРНИТЕ ТЕХНОЛОГИИ

Настоящите технологии могат да предоставят добри възможности за съхраняване на дигиталното аудио и добро представяне при дигитализирането на аудио .

Новите въпроси при проучванията сега са свързани към усъвършенстваните човешки/компютърни приложения, техники и стратегии за контрол над информацията, относно музиката и процеса при създаване на музика. Съвременните техники за извличане на сложна описателна информация от аудио сигналите ще доведе до нови достижения в областта на музикалната материя и контрол над процеса при създаването на музика и представянето ѝ в пространството.

Публичното представяне на дигиталната музика също така ще предостави значителни възможности за проучване. Съвременните подобрения във взаимодействието между хората и компютрите, разбирането на машините и механизмите за интеграция на технологиите с човешките начини за изразяване - ще облагодетелстват тази област с нови възможности за проучвания и търсения. Интеграцията и приложението на съществуващи технологии в контекста на човешкото виждане и изразяване на музиката и интелигентните музикални взаимодействия ще бъде предизвикателство, което трябва да бъде разрешено. Също така има налице значителни възможности за разследване на нови подходи за контрол над компютърната музика, алгоритмичните композиции и приложението на компютърните алгоритми за живо изпълнение и импровизация. [81]

Новите подходи за музикално изразяване ще изискват усъвършенстване и развитие в музикалните възпроизвеждащи системи, като например обкръжаващия звук. Това поле ще се развива, за да включи употребата на пространствени стратегии, които изискват много голям брой от аудио канали и голям брой високоговорители , за да се използва в различни контекстуални варианти (включително в киното и концертните зали). Технологичните подобрения и проучвания във високоговорителните системи, стратегиите за кодиране на пространствения звук и

човешкото усещане за звука – ще дава допълнителна информация за усъвършенстването на технологиите в тази област.

Музиката, изпълнена и представена от компютрите, или генерирана от тях, трябва да бъде показвана, за да предостави възможност на публиката да я разбере или да бъде включена в живите изпълнения и импровизации. Техническите усъвършенствания в тази област ще даде предимство и на близките с музиката области – като естетиката и артистичните проучвания. Новите методи и похвати за изпълнения, местата за изпълнения (например в интернет, публичното пространство или персоналните аудио системи) могат да бъдат доразвити чрез бъдещи технологични проучвания и разработки.

Проучванията в областта на създаването на музика и взаимодействията между технологичните приложения и връзката човек/компютър (*human/computer interface (HCI)*) за създаването ѝ – е установено пространство, което открива музикални приложения и други области от HCI проучването.

Проучванията над компютърните междинни комуникации също е важна област. Откритията в тези области ще продължат, свързани с напредъка между възможностите на електронния хардуер и усещанията в човешките действия, както и с прилагането на сетива-данни за създаването на музика в дигиталните процеси.

2.1.4 НОВИ МЕТОДИ И ПОДХОДИ ПРИ СЪЗДАВАНЕТО НА КОМПЮТЪРНА МУЗИКА

За технологичните нововъведения е подходящо да бъдат приложени в много музикални стилове. Новите форми на дигитална и пост-дигитална музика (чиито действащи области са извън академиите), съвременните джаз, поп, рок, фюжън, етно музика и много други стилове – са облагодетелствани за проучвания в приложението на новите дигитални музикални технологии. Ангажирането с нови музикални форми ще позволи на технологичните проучвания да предоставят на света жизнена и усъвършенствана музикална култура.

Комуникацията и сътрудничеството между артисти и инженери може да предостави безценни резултати в изследванията. По-мощната интеграция между дисциплините и по-малко скованите дефиниции на субективните области – се изискват за придвижване от представите за мултидисциплинарната работа – до концепцията на проучването, което е наследено трансдисциплинарно. Важно е да се установи диалог и

разбиране между творците и техническите откриватели в музиката. Проучванията и обучението в тази област е важно условие за поддръжка на проучванията за бъдеще .

Изследванията в областта на човешкото възприемане на звука и други човешки научни дисциплини имат потенциала да повлияят значително върху създаването на дигитална музика и свързаните инженерни търсения. Дигиталните музикални проучвания трябва да обединят изследователите, работещи в тези области. Забележимо, звукът се представя като най-значимия елемент в развитието на моделите на познавателни операции и има възможността да проникне в същността на проучванията във функциите на мозъка. Важната информация свързана с човешката психология, познания и науката за мозъка, може да произлиза от трансдисциплинарното проучване в тази област, в допълнение към човешкото научно познание, което ще даде познание и за музикалното технологично развитие.

Новите методи за създаване на музика, генериране и контрол е сърцето на развитието в дигиталната музика за последните 50 години. Новата естетика, базирана изцяло върху синтетичен или изкуствен звуков материал и контрола над тях, сега са значими интернационални творчески култури. Новите музикални жанрове и новите методи за генериране на звук ще продължат да бъдат ключовата област за изследване и усъвършенстване. [17]

Проучванията се изискват, за да се предостави възможност на творците да изработват музикални данни в цялостен, мулти-параметричен контрол, умерено представен чрез компютърните технологии. Изследването е необходимо в областта на управлението на звуковия файл и приложението на автоматизираната музикална категоризираща система, за да позволи проследяването на интеграцията на информацията – да информира, казва, обработва сигнала или да проследява творческите процеси.

Компютърните технологии ще продължават да подпомагат новите подходи за композиране на музика. Области, като алгоритмично композиране и взаимодействие на изпълнителя с правилата за композиция – ще бъдат неизчерпаеми области за изследване.

Разработките също се очакват в областта на компютърните алгоритми за живо изпълнение. Тези компютърни системи, които ще си взаимодействат решително с музикантите, като ги поддържат и осигуряват творчески капацитет, ще бъдат синтез от идеи между алгоритмична музика, електронни изпълнения на живо, познавателни науки, изкуствен интелект и концертни изпълнения. Предизвикателството е живите

алгоритми да се обогатят със способности, типични за човешкото изпълнение : ярка интерактивност, автономност, иновация, особеност и достъпност.

Новите форми за нотиране на музиката, нейното представяне и показването на информацията относно звука, ще създаде нови инструменти за музикален анализ и връзка със звуковия материал, по време на създаването на музика. Техниките, които се използват при връзката между звука и нотацията, нотите и визуалното представяне – ще предоставят невероятни възможности на композиторите, изпълнителите, звуковите инженери, продуцентите, музиколозите и музикалните анализатори.

Ще бъде наложително и много важно да се търсят и изследват нови инженерни въведения, технологии и методи в същността на професионалната и съвременна музика. Новите технологии за композиране и аудио обработка трябва да предоставят професионално ниво на музикалните експерти. Предизвикателството за проучанията в областта на дигиталната музика ще бъде да завладее професионалните създатели на музика с различни вариации от иновационни музикални стилове. Това ще интегрира знанието, акумулирано от музикантите относно същността на музиката, достъпността и методите за музика, нейното създаване, разпространяване, изпълняване и интерпретация.

Творческите резултати могат да бъдат доразвити, като илюстрират множеството инженерни търсения в тази област. Това си има своите потенциални предимства : може да покаже същността на проучванията, приносът за бъдещите инженерни проучвания, обществената нагласа за новите стилове в музиката, разпространението и бизнесът да вземе под внимание развитието на научните търсения.

Търсенията в областта на дигиталната музика трябва да се опитват да достигнат до по-широк кръг от музикални общества, където музикалните иновации са добре приети. Предоставянето на възможност за развитие на компютърната музика е ключовата цел в изследванията в тази област. Създаването на нови музикални стилове и начини за изразяване на музиката чрез новите технологии ще позволи на дигиталните музикални проучвания да направят значителен принос за творческата култура и индустрия на 21 век.

2.1.5 ПАРАМЕТРИЧЕН КОНТРОЛ ПРИ СИНТЕЗ НА МУЗИКАТА

През последните десет години бяха създадени множество мощни алгоритми и системи за синтезиране на звука, обработка на дигиталните сигнали и композиране. По

принцип, в основата си тези алгоритми са много сложни и комплексни (те имат изумителен брой от едновременни контролни променливи), а може и да не са толкова интуитивни (например, техните променливи да не са обективно смислени). Това ограничава използването на такива алгоритми и системи.

Като пример за илюстриране на проблема се разглежда един типичен аудио еквалайзер. Той осигурява голям набор от фейдъри за модифициране спектъра на звука. Нормално, един еквалайзер се използва за модифициране на слуховото качество на музиката, за да достигне специфично качество на тона. Налице са голям набор от фейдъри, по-прецизен контрол в еквалайзера, което води до по-усложнени операции, например, нараства броят на възможните комбинации за позициите на фейдърите. Съвременното състояние на аудио технологиите може да предостави високо параметрично ниво на еквайлера с малко повече контролери, всеки един за отделен и специфичен слухов отговор - една степен за плътност, друга степен за точност на звука. Това не са еквивалентни примери за позициите на фейдъра, но правилните контролери дават съдържателни вариации в поставената структура. Осигуряването на високо параметричен контрол над звука и музиката е важно изследователско предизвикателство. Ключът на прогреса е да се развиват ефективни методи, които да интегрират бързо развиващите се аудио и музикални технологии с нарастващите търсения в областта на слушателското и музикално усещане. Това може да бъде постигнато чрез интегриращо изследване и нововъведения от областите на Психоакустиката; Психологията на музиката; Слушателската неврология; Неврологията на музиката; Акустиката; Обработката на аудио сигналите; Музикален анализ.

В много приложения от Акустиката и обработката на аудио сигналите, е необходимо да се разбере какво в действителност чуват хората и как точно се случва това. Звукът, който се състои от звукови вълни под налягане, може акуратно да бъде измерен с усъвършенствано оборудване. Разбирането как точно тези вълни се получават и се организират в човешкия мозък не е процес, който може да бъде пренебрегнат. Област от Психоакустиката е посветена на субективното възприемане на звука от човека; това е психологията на акустичното усещане. Но музиката не е единичен и изолиран звук. По-скоро музиката генерално може се дефинира, като композиция от звуци. Това изисква изучаването на субективното възприемане на организирания звук, което е част от Психологията на музиката.

Въпреки че Психоакустиката и Психологията на музиката допринесоха много за разбирането ни относно субективните човешки възприятия за музика, ние все още не знаем какво се случва в мозъка, когато слушаме звуци и музика. Област от науката за Неврологията на слушателите и излязлата наяве област от Неврологията на музиката – започнаха да поставят все по-често този въпрос. През следващите 10 години, напредъкът в изследванията в тези важни области ще подсили усъвършенстването на нова технология за дигитална музика, много по-значителни нови модели за машинно слушане, нови техники за кодиране на аудио сигналите и нови подходи за организиране на музикалния материал. Сегашните предимства в търсенията, относно Неврологията на слушателите, вече започнаха да дават информация за развитието на DSP чиповете, моделирани след работа над слушателските сетива за възприемане на музиката.

Важно е да се подчертае, че такова развитие и разработки не могат да се случат в изолация и затова резултатите са много по-добри, когато трансдисциплинарното проучване обединява психолози, невролози, инженери и музиканти. Това проучване ще доведе до изобретяване на ново поколение от усъвършенствани технологии за: Звуков дизайн, подпомогнат от компютъра; Средства за музикална продукция; Нови звукови методи; Нови дигитални музикални инструменти и средства за изпълнение; Нови потребителски приложения.

Системите за Компютърно подпомогнат звуков дизайн и средствата за музикална продукция интегрират набор от инструменти за манипулиране на семплирания звук и за манипулиране на синтезирания звук. Има на разположение много техники за имплементиране на подобни средства, повечето от които са от значителна важност за звуковия синтез. Такива системи могат да бъдат трудни за работа, защото им липсват добри качества за високо параметричен контрол.

Звукоизвличането използва аудиото, за да предаде информация или да долови данните слухово. В съответствие с природата на слушателската възприемчивост, за време и резолюция на звуковото напрежение, то формира интересни алтернативи за визуализиращи техники и осигурява добро възприемане на временната информация. Високо параметричното ниво за звуков дизайн, снабден с информация от нашите познавателни способности ще повлияе в огромна степен върху методите за звукоизвличане, специално относно слушателското наблюдение и разбиране на комплексни и едновременно случващи се събития.

Дигиталните музикални инструменти са създадени като жестови контролер и звуково генерирана единица. Жестовият контролер е устройството, което предава

човешките усещания към инструмента. Това е когато физическата връзка между изпълнителя и инструмента започне да се осъществява. Единицата за звукова генерация включва методи от звуковата продукция и нейните контролери. Връзката между жестовите променливи и синтезираните параметри очевидно търпи развитие и може да бъде обогатена и доразвита във всяко едно музикално приложение . Един жест може да контролира голям брой от едновременно синтезирани променливи. Изследването върху методите за високо параметричното интегрирано ниво е първостепенно за изработването на нови дигитални музикални инструменти и увеличаването броя на инструментите.

Финализирайки тази тема, примерът за аудио еквалайзер, цитиран по-рано в тази глава е добро илюстриране на приноса на новите изобретения за високо параметричния контрол при потребителските приложения.

2.2 КОДОВО-ГРАФИЧНО ПРЕОБРАЗУВАНЕ НА МУЗИКА ПРИ ПРЕДСТАВЯНЕ НА МНОГОМЕРНИ ДАННИ

С развитието на изчислителната техника и мултимедията все по-съществени са проблемите за представяне на виртуалните данни и модели, заложи в изчислителните машини. Още в зората на компютрите основно информацията се представя във визуална форма (монитори, принтери). С развитието на компютърната развлекателна индустрия и хардуера/софтуера за музициране, втори много разпространен начин за представяне на данни и модели се превръщат компютърните аудио системи.

С появата на виртуалните среди и светове, към стереоскопичните визуални изображения изключително успешно могат да се приложат стерефонични картини, с цел постигане на все по-голям реализъм. Интерес за изследователите представлява поведението на субект (човек) „потопен” във виртуална визуална среда, обогатена със звукова картина. Подобни експерименти въвличат двете основни сетива при хората (зрение и слух). Много интересен аспект е добавянето на инфразвуци и ултразвуци, макар и те да не са доловими за повечето хора.

2.2.1 ВЛИЯНИЕ НА РАЗЛИЧНИТЕ ХАРАКТЕРИСТИКИ ПРИ ПРОЦЕСА НА ВЪЗПРИЯТИЕ

Височината на дадена мелодия или звук може да повлияе положително или отрицателно на слушателя, в зависимост от заложената идея. Ниските тонове създават усещане за несигурност и дори опасност (например музикалното оформление на напрегнати филмови сцени). Дори в природата, преди да се случи нещо свързано с риск за индивида или групата от индивиди, обикновено има затишие.

За постигането на ефект за напрегнато усещане или очакване се използва ускорен ритъм (насечена мелодия, наситена с високи тонове), което също допринася за увеличаване на напрежението. Много често в алармите и сигналите за опасност се използват звуци с много ниски или много високи тонове (пример сирена на кораб или на локомотив).

Чрез използване на ритмично подредени високи звуци може да се постигне много положително и дори мистично усещане у слушателя, каквито са мелодиите на индианците от Южна Америка. Прецизното използване на височините при звуците, в съчетание с подходящ ритъм може да създаде силно положителни или силно отрицателни възприятия.

Силата на звука също може да влияе положително или отрицателно. От състояние на спокойствие, ако силата на звука се увеличи драстично това създава усещане за напрежение и дискомфорт. По същият начин при напрегната ситуация, спокойна и тиха музика води до снижаване на напрежението и релаксация. Поради тази причина днес все по-широко навлиза използването на музикотерапия. Затихването на звучаща мелодия създава усещането за завършеност и движение във времето. Умелото използване на силата на звука също може да създаде усещане за мистичност.

Ритъмът е един от първите подходи за обмен на звукова информация между хората. Нивото информативност идва от повтаряемостта на звука. Дори и в наши дни има племена комуникиращи си с удари на барабани. Чисто хронологично ударните инструменти се появяват преди всички останали, предестващи дори човешката реч.

Интересно направление, което изключително залага на ритъма и повтаряемостта на два или три тона, е така наречената минималистична музика, която води началото си от 60-те години на 20 век, но наподобява ударната музика на древните африкански племена, в които преобладава постоянният монотонен ритъм. Минималистичната

музика намира приложение в съвременната филмова музика за внушаване на определени усещания у зрителя. Този стил донякъде има и хипнотизиращ ефект.

Влиянието на тембъра е много важна характеристика, която също така може да се използва за внушение. Осезаема разлика се усеща при употребата на различни музикални инструменти. Поради тази причина не е случаен изборът на всеки добър композитор за конкретен солиращ инструмент (флейта, цигулка, обой). Интересно е, че дори отделните инструменти използват различни бленди на звученето си за да подсилат въздействието си над слушателя.

2.2.1.1 КОМБИНАЦИЯ ОТ ДВЕ ИЛИ ПОВЕЧЕ ХАРАКТЕРИСТИКИ НА ЗВУКА И СЪОТВЕТНОТО ИМ ВЪЗДЕЙСТВИЕ

При комбинация на характеристиките височина и ритъм можем да наблюдаваме интересни ситуации. Когато имаме бавна мелодия, наситена с лиризм и хармонични акорди въздействието винаги е положително. За сметка на това, една напрегната мелодия, богата на дисонантни акорди и ускорен ритъм винаги създава усещане за несигурност.

Подобно на мажора и минора, винаги възприятието е положително или отрицателно. Именно поради това, можем да кажем за дисонансни съчетания от акорди, че в тях се наблюдава противовес, несъгласуваност, незавършеност и желание за разрешение. Това е още един белег, който може да се използва за допълване на емоционалното въздействие.

При комбинация от височина, ритъм и сила, когато имаме бавна спокойна музика, която звучи едва доловимо, това въздейства пряко на човека като успокоение. Ако мелодията изведнъж рязко промени силата си тя може да стане дори силно неприятна за слушателя. Добавяйки в тази комбинация и тембъра, спокойната мелодия със солиращ инструмент обой, който се замества впоследствие от солиращ инструмент тромпет или цугтромбон води до усещане за дискомфорт. Важно е да се познават характеристиките на звука, за да може да се постигне максимален ефект от тяхното комбинирание и приложение.

2.2.2 ВЪЗМОЖНОСТИ ЗА ПРЕДСТАВЯНЕ НА МНОГОМЕРНИ ДАННИ СЪС ЗВУК

Ако предположим една виртуална среда богата на аудио-визуални образи и група субекти, на които е отнета възможността за визуално възприятие, то можем да проведем изследване за въздействието на средата върху субектите, които ще реагират единствено на звуковата картина. Същият експеримент може да се проведе с втора група субекти, на които е отнета възможността за слухово възприятие. По този начин може да се направят изводи за ефективността на звука, като допълнително възприятие във един визуален виртуален свят.

Всяка среда за виртуална реалност се характеризира с абстрактен модел на данните. Абстрактният модел на данните присъства в паметта на изчислителната машина и не притежава форма и облик, подходящ за възприемане от човека. За да бъде възприет абстрактният модел за виртуална реалност, се използват похвати за преобразуване и представяне на информацията пред потребителя, във форма подходяща за физическите му възприятия (зрение, звук, допир и т.н.). Ясно е, че визуалните картини ще наподобяват обекти от реалния свят, но възниква въпросът по какъв начин тези обекти да придобият и звукови характеристики, така че потребителят да получава по-висока информираност за абстрактния модел на виртуалната реалност.

Характеристиките на обектите от абстрактния модел, могат да се разделят на характеристики, които ще бъдат представени на потребителя под формата на визуална информация и под формата на аудио информация.

Базов модел за обогатяване на виртуално представения обект може да бъде просвирването на звуци с определена височина. При този максимално опростен модел височината на звуците ще отговаря за създаване на впечатление за по-добра или по-лоша характеристика на обекта, според казаното по-горе, за влиянието на по-ниски звуци, с учестен ритъм (напрегнато усещане) и по-високи звуци, в умерен ритъм (позитивно усещане). Най-лесен начин за реализация е инцидентно прозвучаване, според насочването на вниманието на субекта.

Втори базов модел може да се изгради на основа силата на звука. Обекти с негативни характеристики могат да се представят с дразнещо висока сила на звука, която инстинктивно да отблъсне потребителя или подходящо умерена сила на звука, която само да привлече вниманието му. Силата на звука, използвана едностранно, без

да се комбинира с другите характеристики, трудно може да създаде достатъчно еднозначна реакция от страна на потребителя.

Ритъмът, като изключително благодарна характеристика на звука, може много широко и много успешно да се приложи за обогатяването на виртуални картини. Един възможен вариант е използването на приглушен непрекъснат ритъм, който да създава усещане за постоянно присъствие на потребителя във виртуалната сцена. Този ритъм може да бъде модифициран по своята сила или повтаряемост, в зависимост от това дали потребителят концентрира своето внимание към обекти с по-добри или по-лоши характеристики. Неусетно от обикновен потребител на виртуална реалност потребителя се превръща в композитор на различни звучащи ритми, които многократно биха подсилили илюзията за по-пълно участие на субекта във виртуалната сцена.

Тембърът също се явява много благоприятна характеристика на звука, за „маркирането“ на относително малки разлики в качествата на група сходни обекти. Човешкото ухо и центровете на мозъка, отговорни за обработката на звуците, са силно чувствителни към промени на тембъра, тъй като те служат за различаване на обекти с относително малки разлики в реалния живот (примерно разпознаване на човек по телефона, на база единствено разликата в тембъра на гласа му).

Интересна възможност е виртуалното пространство да бъде осяно с аудио икони. Ако си представим, че потребителят може да манипулира пространството изградено във виртуалната реалност, то при активиране на аудио икона може да получава информация за позицията, която е заел в рамките на виртуалното пространство. Примерно, ако стъпил в забранена зона да прозвучи звукова икона за забрана. На принципа на аудио иконите може да се имплементира техника позната в компютърните военни стратегии с названието „*fog of war*“. Това ще рече, че ако потребителя се е доближил във виртуалното пространство до обект, който може да представлява определен интерес в „търсенето“ на този потребител, то попадането в зоната на обекта да води до изпълнение на звукова икона, с цел привличане на вниманието. Този ефект за привличане на внимание може изключително силно да бъде акцентиран, ако системата за виртуална реалност е снабдена с *Dolby surround* аудио система (хардуер и софтуер).

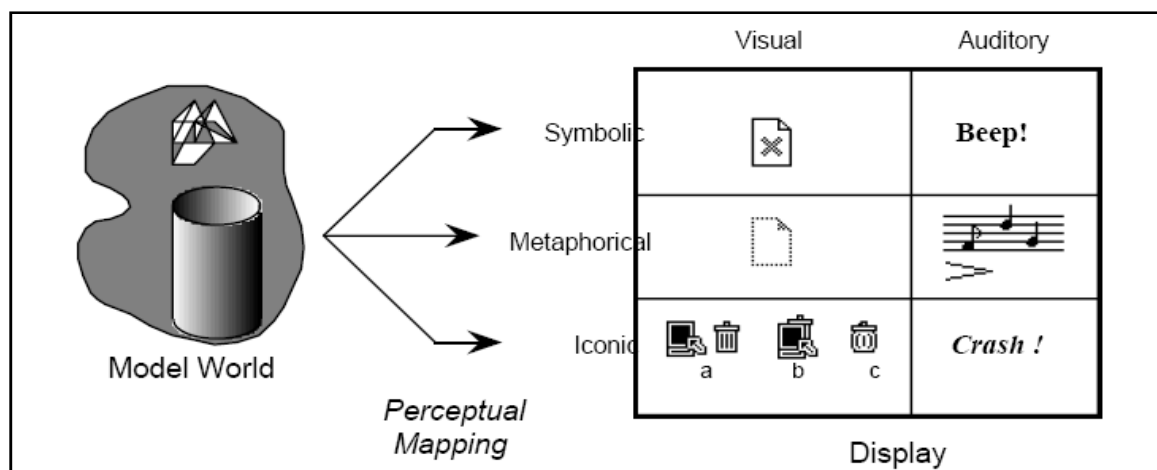
Значително по-усложнена аудио информация може да се предостави на потребителя, ако му се позволи той сам да маркира аудио обекти от виртуалния свят, който разглежда. По този начин потребителят се превръща от пасивен участник в

„аранжор” на сцената от която е неизменна част. Подобна функционалност може да постигне дори символична схема за композиране.

2.2.3 КОДОВО-ГРАФИЧНО ПРЕОБРАЗУВАНЕ НА ЗВУЦИ КЪМ СЪБИТИЯ

Двете дефинирани черти на слуховите икони са употребата на заобикалящи звуци, контролирани чрез параметри на събитията, които ги причиняват и употребата на интуитивни преобразувания между звуците и компютърните събития, които индикират. [24]

Има три различни вида на кодово-графично преобразуване, които позволяват на моделирания свят да се изобрази на монитора: символно, метафорично (образно) и изобразително (Фиг.5).



Фигура 5. Видове кодово-графично преобразуване [132]

Символното кодово-графично преобразуване е трудно за изучаване, защото е изцяло условно и разчита на социалното установено възприемане за неговото доказване.

Метафоричното кодово-графично преобразуване е по-лесно за изучаване, защото разчита на приликата и сходството между представянето и обекта, който ще се изобрази.

Кодово-графичното преобразуване към икони е на-лесния наин за изучаване, защото разчита на сходството между представянето и обекта, който се представя: в случай на възприятийно кодово-графично преобразуване, това означава, че представянето трябва да изобрази (или да озвучи) събитието, което представя. Трябва да се отчете факта, че тази прилика не е субективна величина, но много зависи от физичното преобразуване между физичната структура и възприетото проявление в

ежедневния свят. Кодово-графичното преобразуване към икони позволява на участниците да приложат способностите, които те вече са придобили в ежедневието си към новия моделиран свят, създаан от компютърните технологии.

От тази гледна точка, дизайнерите на звуковите интерфейси трябва да имат предвид кодово-графичното преобразуване между звуци и събития, които да са колкото се може по-реални. Трябва да отбягват създаването на нов символен език за звуци и в замяна на това да се стремят към метафорично или иконно кодово-графично преобразуване.

2.2.4 ОБОБЩЕНИЕ НА СВЕТОВНИТЕ ПРАКТИКИ

Слуховите икони имат огромен потенциал като стратегия за създаване на информативни, интуитивно-достъпни, ненаатрапчиви слухови интерфейси. Те вече показаха способността си да повишават конкретността и да съдействат при директното ангажиране с моделирания свят, създаден в повечето графични интерфейси. Те могат да предоставят обратна връзка относно действията на потребителите и съответно преположения за това какво действие може да последва.

Изключително интересен пример в тази област се открива в работата на група изследователи от Университета в Джорджия. Тази група се ангажира с направата на звуков интерфейс за слепи потребители, който ще транслира информацията и стила на изобразяване, използван в повечето графични интерфейси, към слуховата сфера. Тяхната разработка предполага комбинирането на слухови икони и пространствен звук, което да създаде виртуална среда за слепите – една изкуствена среда, в която хорага могат да си взаимодействат без наличието на графични компоненти. [96]

Разработването на слуховите икони е вече широко разпространена интерфейсна техника , която предполага наличието на мощни компютри с голяма памет, като в допълнение е нужно хардуера да включва възможността да записва и просвирва звуци под софтуерен контрол.

Нарастващата мощ на съвременните компютри дава възможност за създаването на синтезиращи алгоритми, специализирани към генерирането на параметаризирани ежедневни звуци за употребата им като слухови икони.

Звуците имат потенциала да бъдат структурирани така, че да представят многомерни данни. Звукът и визуалните обекти могат да се допълват взаимно при предоставянето на информация. Технологиите за възпроизвеждане на звук са способни

и на разположение, а също така е важно, че вече има много натрупано познание относно естеството и възприемането на звуците. Предизвикателството пред специалистите е да интегрират тези познания и технологии в полезен и употребим компютърно-човешки интерфейс.

Съществуват събития в компютъра, които не могат да се преобразуват към конкретни ежедневни събития. И тук идва въпроса кога ще е полезно да се чуе звук? Несъмнено повечето музикални характеристики могат естествено да се преобразуват към параметри от ежедневиия свят – например, по-висок тон означава качване, а по-нисък тон означава слизване.

Интересна област на изследване е изучаването на връзката между звука и графиката в интерфейсите. Може да се наблюдава как звуците работят заедно с други събития, които предоставят допълнителна информация в потребителския интерфейс.

Взимайки под внимание експанзията на научната визуализация, използваща графиката с присъединени звуци, то тези преобразуващи системи на данни към звуци – ще се превърне в стандартна опция в много от графичните програми.

Взимайки предвид гореизложените изводи можем да дадем и друга насока за прилагането на принципите на фракталния модел в създаването на музикални композиции.

ПРИНОСИ

Във втора глава от дисертацията е засегната основно областта на проучване свързана с компютърната музика и новите подходи за нейното създаване. Дадени са конкретни определения свързани с понятието компютърна музика. Направен е анализ на новите похвати, които се използват при създаването на компютърна музика, измежду които са дигиталния синтез на музиката и параметричния контрол при синтеза на музиката. Дадена е важна концепция за създаването на дигитална музика чрез компютърните технологии.

На базата на направения сравнителен анализ става ясно, че съвременното изкуство вече граничи и се определя от дигиталността и от това доколко се адаптира към приложението и усвояването на компютърните технологии за генерирането му. За повече яснота относно термина дигитален в тази глава също така е направен обстоен анализ над дигиталния синтез на музиката, синтеза на звук, новите методи при създаването на компютърна музика. Друга важна част от изследователската работа в тази глава е частта, която засяга кодово-графичното преобразуване на музика при представяне на многомерни данни и изследване влиянието на различните характеристики при процеса на възприятие.

Днес при наличието на семпли, синтеза на базата на семпли и възможността за бързото изчисляване на по-големи цикли, може да бъде отдадено повече внимание на структурата на композицията.

Установявам че с увеличаване мощността на компютрите стана възможно решаването на задачи с нарастваща комплексност. Съответно с течение на времето композиторите започват да използват компютъра по начин все по-близък до този на творческия процес. Това е неизбежна и жизнено важна тенденция за утвърждаването на алгоритмичната композиция и изобщо в изкуството, създавано чрез компютри. Колкото по-автентично се пресъздава творческия процес, толкова повече компютърът се превръща в средство за реализация на артистичните идеи и все по-малко заместител на вдъхновението.

ГЛАВА 3. ФРАКТАЛНОСТТА КАТО МОДЕЛ ПРИ КОМПОЗИРАНЕТО НА МУЗИКА

Както в древността, така и в наши дни продължават да съществуват опити да се докажат връзките и взаимоотношенията между музиката и математиката. И двете са толкова различни, същевременно толкова еднакви. До ден днешен не спират опитите да се открият тези съотношения, които се намират както между тоновете и техните честоти, така и между честотите и техните числови еквиваленти. Още една страна на изследване е връзката между музиката и цветовете и еквивалентното съдържание между едното и другото. Редица изследователи и творци в тези две области търсят тази връзка и доказват нейното съществуване, като оприличават тоновете на определени цветове. Възприемането на един звук или звуково съчетание като свързано с някакъв цвят наричаме фотизъм или цветно чуване.⁴⁴ Не случайно опитите в областта на електронната музика се съпътстват с подобни еквивалентни образци, като визуализирането на музиката в определени цветове и светлина. Това разбира се, не е случайно, а е продиктувано най-вече от стремежа на творците да провокират всички сетива на публиката и те да възприемат музиката, не само слухово, но и визуално.

Това което най-цялостно и завършено визуално представя тази взаимовръзка са фракталите и техния подмодел - фракталната музика.

3.1 МОДЕЛНОСТ НА ФРАКТАЛНАТА МУЗИКА

Фракталната музика, като обект на изследване може да бъде разглеждана като два модела.

Първият е нейната естествена връзка и отношение спрямо фракталите, като геометрични обекти и нейното интерпретиране спрямо уравненията, които я генерират. Всичко относно този модел и връзките му с геометричната природа на фракталите, както и създаването на алгоритмична музика на базата на тази себеподобност, ще бъде разгледано в следващата глава на настоящата дисертация. Фракталната музика е частен случай на алгоритмичната музика, която е много по-богата на възможности, които предлага.

⁴⁴ Цветно чуване са имали редица музиканти, между които Хайдн, Вебер, Лист, Берлиоз, Сен-Санс /2.с.63/

Вторият модел е този, към който ще насочим своето внимание в тази глава от дисертацията. Той е този при който музиката, правена и композирана, като образец на фракталните обекти, следва тяхната основна черта - себеподобността. Тази област на изследване провокира много от композиторите да търсят и следват тази черта в своята музика. Други творци създават своята музика - следвайки тази себеподобност интуитивно. Тази форма на креативност и отношение към фракталите, съществува от времето на Бах. Никой не може да потвърди или да отрече дали Бах е осъзнавал тази подобност и съзнателно се е стремил към нея или наистина фракталността съществува в съзнанието и гения на творците, така както съществува във всичко около нас. Така както не можем да кажем категорично дали наличието на златната среда в множество творби на бележити композитори, е било използвано съзнателно или несъзнателно, по-същия начин що се отнася до връзката между фракталите и моделите в музиката – те могат да бъдат приети като закономерност. Повече за връзката между музиката на Бах и фракталите, може да се намери в Приложение 2.

Истината е че както повечето необясними и причудливи форми на съществуване около нас, така и фракталите са нещо, което съществува, може да бъде обяснено с научни термини и да бъде приложено в множество области на технологиите, но не може да бъде доказано от къде идва, накъде е тръгнало и докъде ще стигне. Защото то съществува извън нашите представи, но хората никога няма да спрат да се опитват да го анализират и изследват.

Говорейки за музиката като за едно от най-необяснимите и причудливи форми на човешкия гений, също не бихме могли да дадем една точна формула как тя се прави, накъде е тръгнала и има ли край. Въпреки желанието на изследователите да сложат рамки около музиката и да я направят нещо закономерно и константно, тя винаги успява да промени своята форма и начин на изразяване, именно за да докаже своята безкрайност и необхватност.

Това което искам да подчертая и да изведа като факт в тази глава от дисертацията е подобността и сходността на музиката с фракталите, поради тяхната идентичност, в заложената и неизменна форма на безкрайност, повтаряемост и себеподобност. Моделът на фракталната музика не следва изцяло модела на фракталите, но вглеждайки се детайлно в структурата на дадено произведение и разглеждайки детайлно структурата на даден фрактал, ще достигнем до един и същ извод - че себеподобността, повтаряемостта и безкрайността на структурата е налице и в двата модела.

Разбира се нито фракталите наподобяват музиката, нито музиката наподобява фракталите, но се оказва, че две различни форми на изкуство (едното геометричен обект, другото звук) са част от едно цяло, наречено фактор на креативността и заложен алгоритъм за създаване, който не е написан в законите на креативността. Той съществува, за да създава и всяко едно подразделение на изкуството и неговите форми следва този алгоритъм по един неизписан и неустановен път на осъществяване. Този алгоритъм контролира процесите на креативност около нас, във всички сетивни форми, които ни заобикалят.

Този фактор на креативността може да бъде наречен с множество имена, но за мен е ядрото, което създава цялото - това е силата, която движи всички процеси във Вселената и причината за съществуването на видимия, физически свят. Въпреки опитите на учените и изследователите да определят това явление, то си остава една неопределена величина и до ден днешен. Това което наистина бихме могли да направим е да се опитаме да разберем тези явления и форми на креативност, да ги изследваме и да се опитваме да следваме техния алгоритъм и начин на работа, след което да приложим това познание спрямо технологиите и формите на изразяване, които използваме.

Музиката никога не е била възприемана или определяна, като форма и модел на фракталите. Тя винаги се е приемала и изследвала като самостоятелна форма, част от основните форми на изкуството. Отскоро появилите се опити за изследване в областта на музиката, като модел на фракталите, даде основа за търсенето на сходни черти и поведение между музиката и фракталните обекти.

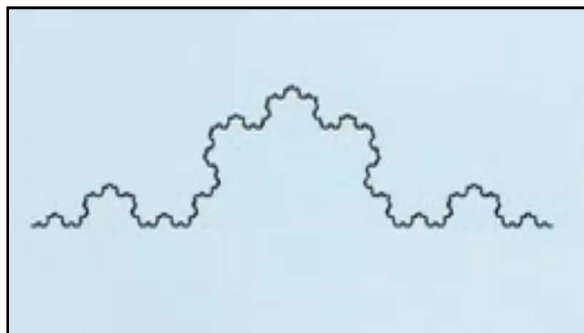
3.2 КРАТКА ИСТОРИЯ НА ФРАКТАЛИТЕ

За да обясним музиката като модел на фрактал по-долу е дадена кратка история на фракталната същност.

Началото на историята на фракталите започва в Германия, пре 1877 г., с математик на име Георг Кантор. Той решава, че ще вземе една права и ще изтрие средната ѝ третина и вземайки двете получени прави ще повтори същото действие, с което ще приложи един рекурсивен процес. Така той започва с една права, след това, после четири, шестнайсет и т.нат. Повтаря този процес безброй пъти и получава безкраен брой прави, всяка от които съдържа безкрайно много точки. Така той осъзнава, че има множество, чийто брой на елементите е по-голям от безкрайността.

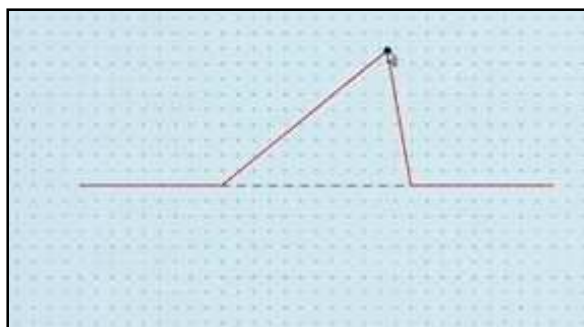
Кантор основава теорията на трансфинитните множества. И други математици правят същото нещо. Шведският математик фон Кох решава че вместо да изважва прави, ще ги добавя. [82]

И така получава тази красива крива. (Фиг.6)



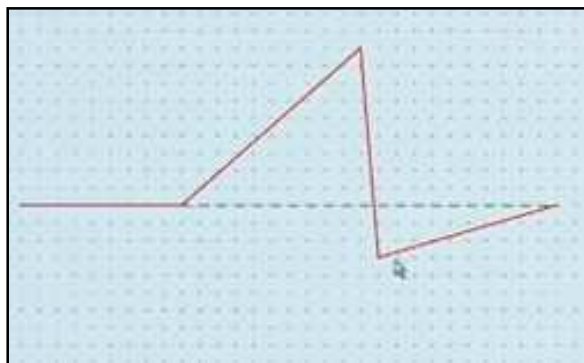
Фигура 6. Снежинката на Кох [82]

Няма конкретна причина защо трябва да се започне с тази начална форма (Фиг.7). Може да използваме която и да е форма.



Фигура 7. Начална фигура [82]

Нека пренаредим тази фигура и издърпаме единия ѝ край някъде надолу (Фиг.8).



Фигура 8. Модифицирана фигура [82]

При итерацията тази начална форма се разгръща в съвсем различно изглеждаща структура (Фиг.9).



Фигура 9. Променена фигура след итерация [82]

Всички тези фигури притежават свойството самоподобие: частта прилича на цялото. Това е същият модел в много различни мащаби. Математиците по онова време смятат, че това е много странно, защото когато се намалява линията, всъщност се измерва все по-голяма дължина. И понеже те минават през итерациите безброй много пъти, както линията се смалва до безкрай, така и дължината стига до безкрай.⁴⁵ Математиците не откриват смисъл в това, наричат тези криви патологични и забравят за тях. Изминават 100 години, когато през 1977 г. Беноа Манделброт, бащата на фракталите, осъзнава, че ако се направи компютърна графика и се използват тези форми, които той нарича фрактали, се получават формите на природата. Природата има тази самоподобна структура. Природата използва самоорганизиращи се системи. [83] Фракталите притежават самоподобие – това че те са подобни на себе си, на не са задължително подобни един на друг. [84]

3.3 ПОВЕДЕНЧЕСКИ МОДЕЛ И СХОДНОСТ МЕЖДУ МУЗИКАТА И ФРАКТАЛИТЕ

Трудно би било да се даде точно определение за поведението на музиката. Тя е необхватна, неизмерима или сравнима с други величини, освен със самата себе си. Но целта на тази глава от дисертацията е именно откриването на такъв поведенчески модел в музиката и сравняването му с поведението на фракталите.

Взимайки в предвид основната характерна черта на фракталите да бъдат себеподобни и безкрайно сходни и еднакви, въпреки своята размерност в детайлите, е хубаво да изградим една матрица и за самата структура на дадено музикално произведение.

⁴⁵ Точно както е казал самият Георг Кантор, *“Рекурсията продължава завинаги”*

Нека се опитаме да съотнесем музикалната форма към прости музикални форми, към прости психологически принципи и да формулираме задачата на композитора според тях.

3.3.1 СТАНДАРТНИ ФОРМИ

Така наречените “стандартни” форми са просто конструктивни модели, но те подлежат и на математически анализ, който има връзка с това изследване. В такъв смисъл “стандартните” форми са полезни категории, които решават формовите проблеми. В тази глава ще разгледаме някои от стандартните форми, за да видим как те прилагат нашите принципи, като целта ни е да изследваме как конкретните модели изразяват общите принципи на формата. **Отделната фраза** демонстрира всичките основни елементи, съставлящи удовлетворителна конструкция. Фразата трябва да има начало, провокиращо интерес. Трябва да стимулира все по-дълбоко участие на слушателя и да създаде чувство за разрешение в края си. Степента на завършеност, дадена от пунктуацията ѝ зависи от позицията на фразата в рамките на произведението. **Периодът** съдържа две фрази, свързани в взаимоотношение въпрос-отговор. Това взаимоотношение се дължи предимно на каденците – първата е затворена, а втората – отворена. **Двойният период** е силно симетрична и затова стабилна и предвидима структура, която също натрупва сила постепенно. Разликата е, че тук напрежението се простира върху четири фрази. Трите вътрешни каденци са подчинени на финалната, която създава пропорционално по сила освобождаване. Двойните периоди са полезни при представяне на нов материал, защото структурата е стабилна и спомага за лесно запомняне. **Фразовата група** е поредица от свързани фрази, без ясната симетрия на период или двоен период. Въпреки това, финалната каденца остава най-силната в групата. Затова фразовата група е по-малко предвидима от периода или двойния период, но все пак изразяваща ясна йерархия в каденцова структура. **Фразовата верига** избягва йерархичното подреждане на каденци. Често последователните ѝ фрази се базират на различен материал. Обикновено се използва в преходни или развиващи секции, тъй като структурата се развива бързо и непредвидимо. Фразовата верига дава конструкционен контраст спрямо периода и двойния период. **Секвенцията** попада в един специален, симетричен вид фразова верига. Секвенцията представлява повторение на една и съща фигура в различни тоналности или ладове. Тя създава инерция, която

евентуално трябва да бъде разчупена от асиметрия. Секвенцията използва принципите на прогресията, и в този смисъл е предвидим похват. **Класическата техника за изграждане на вариации** се основава на разликата между основната структура и орнаментацията. При композирането на класически вариации, основния хармоничен контур и фразовата структура на оригиналната тема остават същите, а нови мотиви и акомпаниментни фигури осигуряват нов материал на повърхността. Оригиналната мелодия би могла да осигурява мелодичен скелет за вариации. Тъй като вариационната форма като цяло е периодична, цялостният дизайн може да стане твърде предвидим. Един начин за избягване на това е създаването на групи вариации в прогресии – няколко последователни вариации биха могли да ускоряват последователно или да водят до съгъстяваща се структура. Обикновено последната вариация се отделя от останалите, като излиза от общата основна структура. **Простата третична форма (ABA)** се базира на един основен структурен принцип: промяна, последвана от завръщане на предишния материал. В тази най-проста форма, завръщането е буквално. Предвидимостта, която се поражда от това, прави формата подходяща за малки движения с лек характер – например танцовите движения в класическите симфонии. По-драматичните идеи изискват по-сложни форми за да създадат повече структурно напрежение. В простата третична форма основната и средна секции са конструирани като затворени форми – всяка от тях има напълно завършена каденца и може да бъде чута и самостоятелно. Средната секция има свой мотив и, ако хармонията е тонална, е в близка тоналност. **Двоична форма** - има много вариации на двоичната форма: двете части могат да бъдат или да не бъдат симетрични, първата може да има завършена каденца (“секционна двоична”) или отворена (“непрекъсната двоична”); втората част може да връща части от материала в първата. Типично за формата е, че двете секции разработват един и същи материал. Обикновено първата част е период или двоен период. В по-простите типове (симетрична, секционна), основният тип контраст във втората част е в хармоничните детайли. В по-сложните началото на втората секция може да действа като миниатюрно развитие. Структурата му е по-малко стабилна и предвидима. Често се използват секвенции. **Сложната третична форма** въвежда ясна йерархия: всяка секция е сама по себе си двоична форма (обикновено непрекъсната). Това създава ефект на разширение и обогатяване на цялостната форма. **Сонатата** е развитие на двоичната форма. Драматизмът и богатството, характерни за тази форма, произлизат от обхват – значителна продължителност, съдържаща силно контрастиращ материал; стабилни начални представяния на материала, свързани от преходна секция;

дългосрочно напрежение, породено от слилна отворена пунктуация в края на втората тема. Сонатата е сложна, богата на напрежение разказвателна структура, със сериозен потенциал за отклонения, усложнявания и богати баланси. Освен това дава възможност за изследване на материала в различни формови контексти. Сонатата е много полезна в дълги произведения заради присъщото ѝ напрежение. Лесно се адаптира към различни хармонични стилове, защото основните принципи – баланс чрез варирани репризи, контраст и изненада в конструирането на материала – отговарят на психологическите изисквания за поддържане на интерес и емоционален интензитет през дълъг период от време. Сонатата, типично разделена на експозиция, развитие и рекапитулация, може да съдържа и въведение и/или coda. **Рондо соната** функционира като соната, с тази разлика, че секцията на развитието се предхожда от реприз на главната тема, типичен за рондото. Самото развитие играе ролята на втори епизод в нормално рондо. Връщането на главната тема след контрастиращия материал значително намалява напрежението и прави формата подходяща и за случаи, в които се търси по-слаб драматизъм.

Това което най-вероятно привлича вниманието и интереса към поведенческия модел на музикалните форми е тяхното повторение и имитация, което в повечето случаи е заложено предварително като шаблон, който трябва да бъде следван и спазван. Тези предвидими форми на поведение ни помагат да съотнесем фракталния модел към музиката. Именно имитацията и безкрайното повторение са част от изследването, което е представено в Приложение 2, в което се търси връзката между музиката на Бах, фракталите и литографиите на Ешер.

3.4 ПОВТОРЕНИЕТО В МУЗИКАТА КАТО ФОРМА НА ФРАКТАЛЕН МОДЕЛ

3.4.1 ПРОЦЕСНА МУЗИКА

Процесната музика е стил с голямо влияние над музиката през последната половина на 20 век. Основава се на крайно обикновен, минимален музикален материал, със стабилен, пулсиращ ритъм. Музикалният материал се повтаря продължително време и постепенно се развива. Този стил донякъде има хипнотизиращ ефект, разбираем е и добива голяма популярност. Водещите автори на процесна музика са

американските композитори Филип Глас, Стив Райх и Тери Райли. (*Приложение 3, M18M_1976_Reich.mp3*).

Процесната музика се нарича и минималистична, системна и повтаряща се музика. Голямо влияние над този стил музика имат класическата индийска музика, гамеланската музика и африканската ударна музика. (*Приложение 3, New Cities In Ancient Lands, Africa.mp3*)

Филип Глас е един от неколцината американски композитори, който създава творби от прост, строго ритмичен и повтарящ се материал, проектиращ се в дълъг времеви период и често използващ променлив ритъм и тембри. През 1968 г. Глас започва експерименти с минималистични отрязъци от порядъка $1 + 1$, след което създава серия от творби, които съчетават натрупващи се процеси, базирани се на поредица от 1, 1 2, 1 2 3, 1 2 3 4 ноти.

В известен смисъл Филип Глас създава музика, която се доближава и до фракталния модел. По чисто интуитивен начин композиторът се приближава до имажинерното разбиране за алгоритъм. (*Приложение 3, The Kiss; Closing; Escape*)

Използвайки вариации върху три тона и полутоновете между тях, се генерират десетки варианти и комбинации между тях, което много наподобява алгоритъм, който използва случайния и произволен принцип за генериране на ноти.

Зад този математически модел ние сме ограничени в степените на свобода, защото изследваме височината и вариациите ѝ, които възникват при взаимодействието между три степени на лада.

Може да се твърди, че процесната музика е създадена от алгоритъм, чрез математическа формула. Истината е, че няма заложена формула – т.е. процесната музика не използва алгоритъм, въпреки, че много наподобява фракталната музика.

Композиторите на процесна музика не са предполагали наличието на фрактална музика – те композират от началото на 60-те години. Актуални са и до днес.

3.4.2 ПОВТАРЯЕМОСТ В СЪВРЕМЕННАТА ПОПУЛЯРНА МУЗИКА

Човекът който прилага информационната теория от Абрахам Молс (*Abraham Moles*) към музиката, признава ефекта на повторението върху информацията. Той говори за т.нар. “закон на повторение”. [71] Съобразно този закон, колкото повече повторения се случат, толкова по-малко информация се предава. Повторението също

така се използва като критерий в музикалния анализ. Методът за анализ на Николас Рувет включва техника за “процедура на разделяне”, базирана на принципите на повторението. Основна част от теорията на Рувет е емпиричното признаване на огромната роля, която играе повторението в музиката. Оказва се, че повторението е полезен инструмент за откриване на стилистични различия или откриване на нови стилове. Повторението може да бъде разглеждано като емоционален елемент от музиката. В зависимост от броя повторения музиката придобива различен ефект и значение. Важни фактори са функцията и структурата на повторение, както и колко пъти преминава повторението.⁴⁶ Ричард Мидълтън дава няколко дефиниции за термина повторение в музиката: “най-познатият вид повторение са т.нар. *рифове*,⁴⁷ които се откриват в афроамериканската музика и в рока”. [70]

Нека разгледаме една песен, която има симетрична структура. Песента е много популярна и се изпълнява от Елвис Пресли, казва се *Love me tender*. Оригиналът на тази песен е една стара ирландска песен “*Aura Lee*”, която става известна благодарение на Елвис Пресли. Тази песен е в типична баладна форма, която обикновено се състои от 32 такта, докато в този случай тактовете са 16.

Четиритактовата фраза «a» може да бъде разделена на по-малки единици, което ще подпомогне идентификацията на структурата. Симетричната форма работи в групи от по 4 такта. Песента започва с четири тактова мелодична и хармонична фраза «a», която се повтаря без изменение, но с различен текст (4 + 4) «a» «a» (Фиг.10). Това бива последвано от нова мелодична четири тактова фраза и последната фраза е вариация на първата фраза = a a b a’.

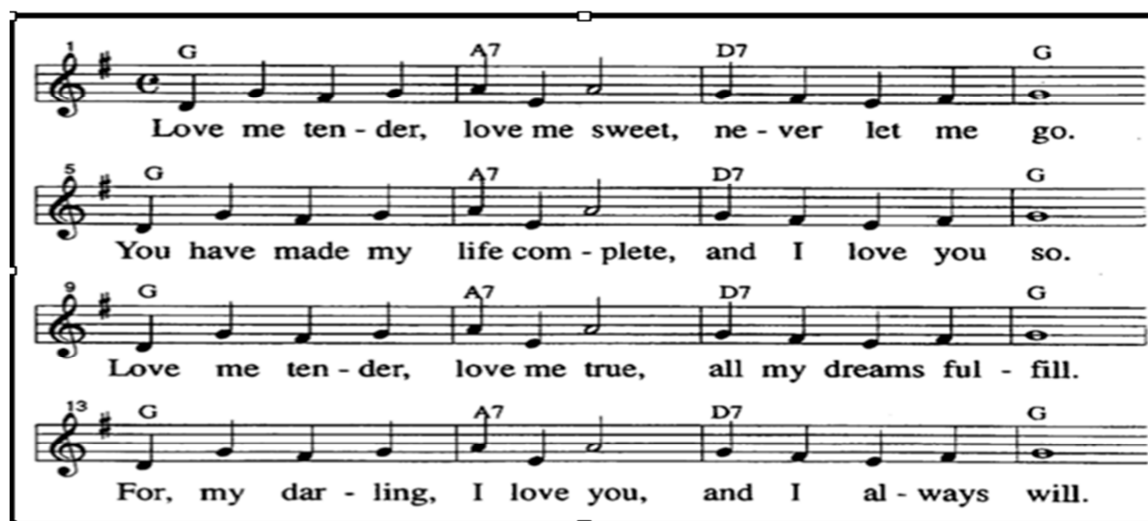
⁴⁶ “Structure and function in musical repetition”, Journal of the Canadian association of university school of music, 1978

⁴⁷ Виж речника



Фигура 10. Четиритактова мелодична фраза от песента Love me tender [133]

Известно е от песента, че след първата фраза, влиза нова. Но какво ще се случи ако това се промени? В следващия пример, структурата на песента се променя, така че фраза *b*, която е контрастна на *a*, се замества с друга мелодична *a* (Фиг.11).

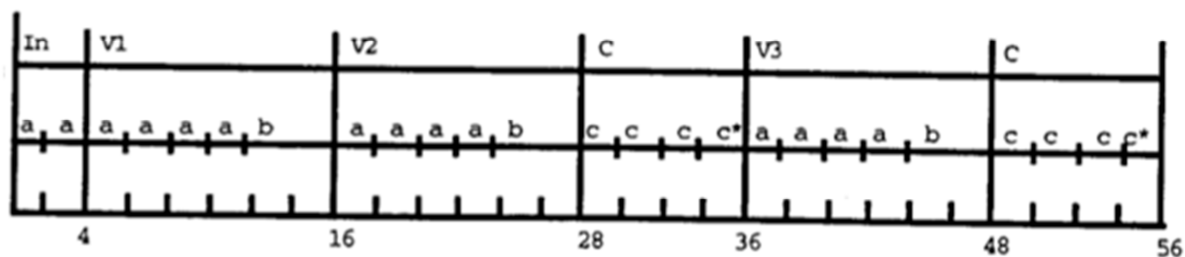


Фигура 11. Променена структура на песента Love me tender [133]

Ефектът от тази промяна е значителен. Структурата на песента сега е просто повторение на $4 + 4 + 4 + 4$ неизменни такта, като всяка четиритактова фраза е идентична с останалите. Песента се превръща в един цикъл от 16 такта. В този пример повторението се среща на ниво фраза.

Нека разгледаме още една, също много популярна песен на Джон Ленън (*John Lennon*) - *Imagine*. Формата на песента е типична форма за популярната музика А А В

А В. Както може да се види от фигурата, куплета А и припева В съдържат малък брой от различни елементи (Фиг.12).



Фигура 12. Структура на песента Imagine [133]

В предишния пример, в песента *“Love me tender”*, повтарящата се четиритактова структура има черти на пълна канденцова форма I/ II/ V/ I. В *Imagine* хармоничната структура е построена от двойки, които сменят хармонията си, т.е. не се наблюдава един пълен хармоничен цикъл (Фиг.13).



Фигура 13. Хармоничен цикъл в песента Imagine [133]

Следващото заместване ще бъде направено за първите 12 такта от песента, които представят секция А. Както се вижда от примера, двутактовата хармония и мелодия се базират на два акорда I (-⁷)/IV. това се повтаря четири пъти. След това следва четиритактова хармонична прогресия IV-VI/ II⁷-IV/ V-I/ V⁷. Интересна замяна тук ще бъде да се промени броя на повторенията на двутактовата структура и след това да се наблюдават възможните ефекти от променената структура. Променените параметри са мелодията и хармонията. В първия заместващ случай двутактовата акордова прогресия

се повтаря шест пъти, вместо четири. Примерът е без текст, понеже са добавени повече тактове към мелодията (Фиг.14).



Фигура 14. Първа заместена версия на песента [133]

Както може да се забележи, добавяйки тези два такта преди четиритактовата прогресия, не нарушава много структурата. Ефектът е слаб, въпреки че оригиналната структура е нарушена. Промяната ще бъде забелязана от слушателя, поради факта, че песента е много известна и слушателите очакват навлизането на четирите такта прогресия след първите 8 такта. Следващото място, където промяната може да се очаква ще бъде след четирите допълнителни такта (12 такта след началото).

Оригиналната структура на песента е конструирана от $8 + 4$ такта и първата променена версия от $12 + 4$ такта. Наистина голям ефект може да се постигне чрез промяна на структурата така, че да не е възможно тя да се раздели на 4. В следващия пример структурата е конструирана от 3 от тези такта ($2 + 2 + 2 = 6$), които биват последвани от оригиналните 4 такта (Фиг.15).



Фигура 15. Втора заместена версия на песента [133]

В този случай последните четири такта влизат съвсем неочаквано след шестте такта. Симетрията е нагласена за 4 тактова структура. Във всяка една песен симетрията е правило, което бива следвано от конструкцията на тактовете и от ритъма. Ако тези норми се разбият, вниманието на слушателя ще бъде привлечено.

Тук може да се направи еквивалентна връзка с математическата повтаряемост на събитието. Повторението е част от симетрията и от установяването на мотиви. Повторението на дадена мелодия или ритмична фигура помага да се унифицира творбата. Ако обаче фразата или фигурата се повтарят твърде често, това може да отегчи слушателя. Музиката се основава на повторения. Музиката въздейства поради факта, че ние запомняме тоновете, които сме чули и ги свързваме с новите, които чуваме. Повторението, ако е направено от способен композитор, може да достави едно огромно емоционално удоволствие на слушателя. Теодор Адорно (*Theodor Adorno*) критикува повторението и популярната музика, като ги нарича инфантилни. Но като негов опонент, Ричард Мидълтън твърди, че ”докато повторението е черта на цялата музика, на всички стилове музика, то едно високо ниво на повторение може да бъде специфична марка на популярната музика. [70] Няма универсална норма за броя или типа на повторението, понеже цялата музика съдържа повторения, но в различна степен и вариантност. Типове повторения включват “точно повторение” (*aaa*), “повторение след отклонение” (*aba* или *aba'*) и “никакво повторение” (*abcd*).

Структурата на музикалните форми на песните в популярната музика типично са секционни, с повтарящи се форми, като строфите. Куплетите и припевите се считат за основни елементи. Обикновено всеки куплет има една и съща мелодия (с някои малки

модификации), но със смяна на текста. Припевите обикновено имат мелодична фраза и ключова линия в текста, който се повтаря. Поп парчетата могат да имат интродукция и кода, както и преход, който както се разбира и от името си, е секция от песента, която свързва куплета и припева в една или повече точки в песента. Повечето песни, най-вече рок и блус песните, имат соло секция. В съвременните стилове музика, влиянието и важността на повторението продължава да расте, защото успеха на една песен до голяма степен зависи от повторемостта на дадени елементи в песента, благодарение на което тази песен се запомня. [71] Има и друг парадокс, свързан с повторемостта - благодарение на медиите, които в днешно време контролират успеха или неуспеха на дадена песен, ако все пак решат да направят една песен, която се състои от 5 тона - в хит, те ще го направят, само като повторят песента, колкото се може по-често. Неколкократното повторение води до промяна на отношението на слушателя. Горепосочените музикални примери доказват наличието на математиката като структурално средство за изследване на музикалните произведения.

3.5 СЪВРЕМЕННИ МУЗИКАЛНИ ФОРМИ И ТЯХНОТО СХОДСТВО С ФРАКТАЛНИЯ МОДЕЛ

Съвременните стилове са сходни със структурата на фракталите по това, че притежават мелодични модели в припевите и заложена ритмическа повторемост. Популярността на днешните стилове музика не би била възможна без приликата им със структурата на геометрични обекти, които са напълно непознати на аудиторията, която слуша тази музика. Слушателите дори не подозират какво се крие зад песните, които харесват и които могат да слушат многократно - те дори не се питат на какво се дължи това. А успеха на съвременните музикални форми се крие в повторемостта и наблягането на даден мелодичен модел, който се повтаря многократно, за да бъде запомнен и идентифициран с песента, към която принадлежи. Очевиден факт е че съвременните песни не блестят с невероятна сложност, разнообразии в хармонията или резки обрати в ритмичната структура. Те са достъпни, опростени, изтъкани от повторения и еднаквост на мелодичните и ритмични модели на всяко едно ниво.

Съвременните музикални форми са един типичен пример за сходство между музиката и фракталния модел в нея. Във всяко едно музикално съвременно произведение, което е част от съвременните жанрове на поп, соул, рок, метъл, е

заложена определена матрица за структурата на тези произведения, която разбира се има свои собствени подразделения и разклонения, но ние ще се запознаем с най-основната схема. Някои типични примери за структура на популярна песен:

Въведение/куплет/припев/въведение/куплет/припев/преход/припев/припев

Въведение/куплет/отклонение/припев/куплет/отклонение/припев/преход/припев/

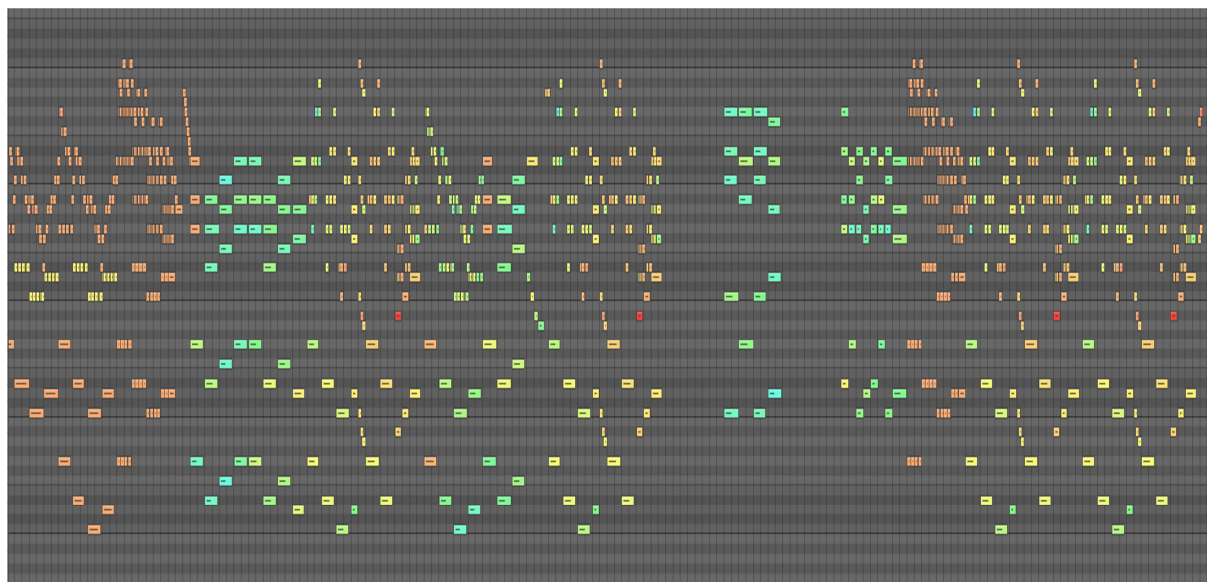
Въведение/куплет/припев/припев/куплет/припев/преход/припев/припев/

Фракталният модел на музиката е един нов, различен начин и метод да се анализира и обхване музикалната материя от хора, които се занимават предимно с визуални обекти. Освен това разбирането, анализирането и правенето на музиката по подобие на фракталните обекти - правено съзнателно или не може да помогне на композиторите и слушателите да открият съвсем нови черти и характеристики на музиката, които не са предполагали, че могат да бъдат отнесени към геометрични обекти. Средствата за това са анализ, структуриране, сравняване и имплементиране на тези похвати както в съзнателния творчески процес, така и в една автоматизирана система за композиция, каквито са софтуерните приложения, създадени за композиране на алгоритмична музика. Неслучайно връзката между музиката и фракталните обекти, музиката и математиката, музиката и цветовете, може да бъде отнесена към хората на изкуството, които докато композират могат да си представят музиката в цветове, да я виждат в образи. Това е възможност, която се наблюдава и при слушателите, които по време на изсвирването на дадено произведение, предпочитат да визуализират музикалното съдържание, което чуват. Това разбира се е един различен начин на възприемане на модела на музиката, като визуализирана картина от звуци.

Трети модел на това да се превежда музиката в образи, е като се използват похватите на фракталната геометрия, които да имплементират резултатите от дадено фрактално уравнение към музикални параметри. Оказва се че за музиката има много средства и методи, чрез които тя да бъде интерпретирана, разбрана, правена, анализирана и превеждана към други форми.

Друг начин да се търси сходство между фракталността и музиката е представянето на музикалния материал на едно произведение в неговия MIDI вариант. Ако се вгледате внимателно в графиката по-долу ще забележите фракталността на

музикалният материал според графичното му представяне. Много лесно може да се забележи къде има повтаряеми форми и как те всъщност представляват графично един фрактал (Фиг.16). А как звучи този MIDI модел на фрактал, може да чуете в *Приложение 3, в композицията ми The crying of the sun.*



Фигура 16. Сходство между фракталите и графичното представяне на MIDI информацията

3.5.1 СВОЙСТВОТО НА ФРАКТАЛНОСТ В МУЗИКАТА

Немалко явления в музиката проявяват свойството фракталност (самоподобие): делението на продължителността; тактовата организация в метричен период от 2-такта, 4-такта, 8-такта; мензуралните канони; някои видове мотивни и серийни преобразувания.

Има категоризирани четири основни начина за използването на фракталността в принципите на композицията, по класификацията на холандския музиковед Нилс Путеманс [23]: конвертиране на графично изображение; по формула - определяне на височините и продължителността на звуците чрез аритметични пресмятания; по генезис - определен музикален параметър се трансформира така, че резултатът от това запълване демонстрира фрактален характер; фрактална композиция чрез игрово поведение - алеаторични структури,⁴⁸ в които изпълнителите създават модел на поведение в сложни системи.

⁴⁸ Виж речника

ПРИНОСИ

В трета глава от настоящата дисертация, озаглавена “Фракталността като модел при композирането на музика”, са изведени интересни факти свързани с историята на фракталите и намирането на естествени взаимовръзки между тяхната структура и принципи на изграждане и тези, на които се основава създаването на музика. Изтъкнати са важни черти на музиката, които имат сходство с фракталността и се прави директна асоциация между музиката, като фрактален модел и самите фрактали. Посочени са основните музикални форми, чрез които отново се търси съпоставка между фракталите и музиката. Също така се показват и няколко примера за съвременната музика и нейната връзка с фракталността. Набляга се и на един основен похват, който се използва както при създаването на фракталите, така и при композирането на музика и това е повтаряемостта. Един похват който успешно прилага фракталност в музиката и я прави уникална и незабравима.

Всяко изследване, което засяга фракталите не може да се осъществи без използването на мощен математически апарат, при наличието на множество масиви от бази данни, обработвани с компютър. Компютрите не са само средство за автоматизиране на процеси, а и инструмент за изучаване на това, което преди не е било възможно да се изучи, например в математиката - поведението на сложни функции и тяхната визуализация. Резултатът от действието на такива функции са най-интересните математически обекти - фракталите.

Доказвам, че немалко явления в музиката проявяват фракталност в структурата си, като едно от основните свойства, които се откриват като сходство между фракталите и музиката са делението на трайността, тактовата организация, лайтмотивните повторения, мотивни и серийни преобразувания.

ГЛАВА 4. АНАЛИЗ НА МЕТОДИТЕ ЗА СЪЗДАВАНЕ НА АЛГОРИТМИЧНА МУЗИКА

В тази глава от дисертацията ще се фокусираме върху употребата на компютрите за генериране на алгоритмични композиции. Това изследване включва най-известните методи, които се използват за алгоритмично композиране - стохастични процеси, вериги на Марков; дву и тримерни хаотични системи; фрактали; алгоритми получени от спектрален шум; клетъчна автоматизация, генетични алгоритми, L-системи. Във връзка с анализа на алгоритмичните методи ще бъдат разгледани пет техники за алгоритмично композиране, базирани на 5 алгоритъма.

Развитието на компютърните технологии през последните години предложи на композиторите начин да обединят тези нови структури в композиционния процес. Тези нови структури са взаимствани от една друга дисциплина – математиката.

Ако сметнем, че има различни методи за алгоритмично композиране, може да се отдаде значителна важност на разграничението между конструкция и резултатна форма, например, между композиционна идея, нейната реализация в музикалния материал и нейното възприемане от аудиторията.

Компютърно-подпомогнатата алгоритмична композиция се състои от конструирани или избираеми алгоритми за генериране на композиция. Тези конструкции се оценяват и определят спрямо това дали получената форма е проработила музикално. Казано по друг начин, компютърното разграничение между алгоритъм и получен резултат — е общоприета форма, която има отличително свойство при компютърно-подпомогнатата алгоритмична композиция. Много рядко в процеса на генериране на композицията, конструкцията на алгоритъма се прави видима.

Теорията на алгоритмичното композиране обикновено се изкривява между партитурния синтез (компютърно-подпомогнатата композиция, обикновено се изпълнява от традиционни акустични инструменти) и звуковия синтез (компютърно-подпомогнатата композиция изпълнява синтетичен звук, който може да бъде чул през високоговорители). Това различие има своите корени в традиционното различие между партитура и инструмент, но компютърно-генерираната цялост между два различни звука, е едновременно партитурен и звуков синтез. И при двата вида синтез, появата на събития във времето е структурирано, като глобално се отнася за формата и локално, отнасящо се за звука и тембъра. Настоящия текст се отнася за по-голямата част от различни методи на компютърно-подпомогнато партитурно синтезиране на оркестрови

инструменти. Не се отнася за по-широката област от компютърно-интерактивни композиции в реално време, нито пък се отнася за алгоритмичния звуков синтез.

Терминът партитурен синтез е близко свързан с компютърната музика, компютърно-подпомогнатата композиция, САО – композиция подпомогната от компютър (*composition assistée par ordinateur*), автоматизираната композиция или алгоритмичната композиция. Алгоритмичното композиране не изисква задължително употребата на компютър. Алгоритмичните процеси също така могат да бъдат показани в конкретни процедури в аналогови и електронни акустични музикални студиа, където те се наричат полу-автоматизирано и автоматизирано композиране

Избора или конструкцията на алгоритмите за музикални приложения могат да бъдат разделени в три категории:

1. Моделиране на традиционни, не-алгоритмични композиционни процедури
2. Моделиране на нови, оригинални композиционни процедури, различни от онези познати преди
3. Селектиращи алгоритми от допълнителни музикални дисциплини

Третата категория получава по-голяма тежест, понеже съдържа най-съвременните подобрения в компютърната музика.

4.1 МОДЕЛИРАНЕ НА ТРАДИЦИОННИ, НЕ-АЛГОРИТМИЧНИ КОМПОЗИЦИОННИ ПРОЦЕДУРИ

Подходът на композитора Леджарън Хилър да формира се характеризира от традиционен начин на мислене. В неговите ранни компютърно-генерирани композиции, чийто брой е измежду най-ранните експерименти в партитурния синтез, той лесно взаймства традиционни форми и използва компютъра, за да прави по-лесен композиционния процес. Но той работи толкова умело с компютъра, че открива нови величини. Първите три експеримента на Хилър не просто представят отделни пиеси, а цели класове от композиции еквивалентни на избраните форми. Неговият четвърти експеримент генерира нови, непредсказуеми форми, за първи път, които директно променят неговото собствено разбиране за формата. От тези експерименти се появява *“Illiac Suite for String Quartet”* за струнен квартет (1955–1956), от Хилър и Леонард Исаксън (*Leonard M. Isaacson*). Четирите части на сюитата документират четирите експеримента, които използват Хилър и Леонард, за да открият как една композиция

може да бъде генерирана от компютър. Първата част (*“Monody, Two-Part u Four-Part Writing”*) програмира 16 различни правила в три категории: какво е позволено, какво е забранено и какво се изисква. Това са правила, които включват и по-лесната полифония. В тази структура, основана на правила, индивидуални музикални събития се определят от случайни процедури. Резултатите са прости *cantus firmus* мелодии с различна продължителност. Вторият експеримент (*“Four-Part First-Species Counterpoint”*) разширява първия, за да постигне различни музикални стилове. Третият (*“Experimental Music”*) е бил програмиран със серийни структури и техники. Методите на четвъртия експеримент (*“Markov Chain Music”*) повлияват на късните работи, които използват компютърно-подпомогнат партитурен синтез.

Хилър и Леонард не спазват в тези експерименти правилата на композиция, но използват редици от зависими произволни параметри. Слушателят може да чуе традиционни музикални форми, но алгоритмите, които са използвани, за да ги генерират остават скрити. Слушателят не може да разграничи дали пиесата е била генерирана, благодарение на алгоритъм или на традиционните похвати на композиране.

Други важни представители на ранния партитурен синтез са Готфрид Кьониг (*Gottfried Koenig*) и Янис Ксенакис (*Iannis Xenakis*). Кьониг започва да разработва своята композиционна програма *Projekt 1* (PR1) през 1963 г. През 1966, той започва работа над *Projekt 2* (PR2). *Projekt 1* описва основни серийни композиционни стратегии, които предоставят различни резултати с помощта на произволен генератор. С *Projekt 2* Кьониг разработва композиция, която трябва да бъде изградена стъпка по стъпка : индивидуални величини за различни параметри на композиционната структура (инструменти, деление на време, хармония/мелодична област, сила); правила за избиране на тези величини и правила за комбинирането им. Използвайки *Projekt 2*, един композитор трябва да си обясни действието на всяка стъпка и да анализира намерението си преди да започне да композира.

4.1.1 МОДЕЛИРАНЕ НА НОВИ ПРОЦЕДУРИ

Гореспоменатият композиционен алгоритъм на Кьониг е замислен да разрешава основни музикални проблеми и да бъде на разположение за употреба от композиторите. Алгоритмите на композитора Кларенц Барлоу (*Clarenz Barlow*) от своя страна, имат характера на частен език и всеки е разработен за отделна пиеса. Тук алгоритмите не са просто композиционни инструменти, а са сами по себе си са едно абстрактно ниво — един аспект от конкретната музикална проява. Барлоу твърди, че ще получи същите резултати без помощта на компютъра и защитава сам себе си срещу термина “компютърна музика.” [12] Неговите числови изследвания на тоналността намират своята изява в пиесата за пиано “*Çogluotobüsisletmesi*”. Няколко звукови пласта преминават едновременно в 14 различни темпа. При смяна на тоналността могат да бъдат чути четвърт-тонови интервали, но не и във всяка тоналност. В тази пиеса “*Çogluotobüsisletmesi*”, метричната и хармонична интензивност, ритмичната и мелодична еднаквост, акордовата плътност, динамиката, акцентите и артикулацията са изчислени съобразно композиционната мета-структура на алгоритмите. Аткуалната пиеса е само една от множеството възможни реализации.

4.2 ИЗБОР НА АЛГОРИТМИ ОТ МУЗИКАЛНИ ДИСЦИПЛИНИ – МЕТОДИ ЗА КОМПОЗИРАНЕ НА АЛГОРИТМИЧНА МУЗИКА

4.2.1 L-СИСТЕМИТЕ (L-SYSTEMS)

L-системите са пряко свързани с „формалните граматики”. За първи път биват изследвани през 1968 г. от Аристид Линденмайер (*Aristid Lindenmayer*), като основа за аксиоматична теория на развитието и като инструмент за реалистично моделиране на живи организми. L-системите са основани на граматиките на лингвиста Ноам Чомски (*Noam Chomsky*). Докато Чомски оформя представянето на изреченията в един език, Линденмайер моделира нарастващите процеси при растенията. Тези системи могат да бъдат категоризирани в два основни типа: контекстно-свободни и контекстно-зависими. Една детерминирана, контекстно-свободна L-граматика се дефинира като наредена тройка

$$G = (A, P, \alpha),$$

където A е азбуката на системата (множеството от всички символи, включително празния символ или нула), P е списък от заместващи правила. Заместващите правила се прилагат към аксиомата рекурсивно, което генерира нова редица от символи, наречени производни низове. Броят пъти, при които се прилагат правилата върху производните низове е познат като рекурсивно ниво.

Контекстно-зависимите L-системи включват стохастични и йерархични граматики, както и параметрични разширения. В тези типове на имплементиране заместващите правила се прилагат, като си присвояват състоянието на обкръжаващите, съседни символи. Позволява по-реалистично и гъвкаво моделиране на организмите. Стохастичните техники внедряват L-системите, за да симулират природните недетерминирани, произволно растящи образци. Тази имплементация преобразува редица от вероятности ($P_1, P_2, P_3, \dots, P_N$) с редица от заместващи правила, включващи символа s като предшественик:

$$P_1: s \rightarrow \frac{1}{4}a; \frac{3}{4}b,$$

което означава, че символа s има 0.25 (25%) вероятност да бъде заместен от a и 0.75 (75%) вероятност да бъде заместен от b . Сумата от вероятности за който и да е символ трябва винаги да е равна на 1 (100%).

Употребата на L-системите е интересна за композиторите по две причини. Първата е, че много често сложните нарастващи процеси могат да бъдат представени от конструкцията на много прости производни функции, които представляват естетическа и предизвикателна опозиция. Втората е, че непредсказуеми, основно себеподобни получаващи се форми нарастват, което не би могло да се постигне от обикновени композиционни методи. Думата себеподобен, станала популярна от теорията на хаоса, играе огромна роля в дискусията около компютърно-подпомогнатото композиране. [32] Всъщност се наблюдава тенденция да се разглежда себеподобността като решение на много формални проблеми. Употребата на композиционни правила, предоставени от L-системите обикновено не е достатъчна за генериране на структурно последователна композиция.

Следващите заместващи правила са прост пример за една такава L-система:

$$a \rightarrow b$$

$$b \rightarrow ab$$

В този пример, a се замества от b и b от ab . Началният символ който започва процеса се нарича *аксиома*. Ако започнем с a , то се получава следната последователност от низове:

a
 b
 ab
 bab
 $abbab$
 $bababbab$
 $abbabbababbab$
 $bababbababbabbababbab$

Ако запишем в редица дължината на получените производни низове, то ще получим редицата на Фибоначи. L-системите са популярен избор на композиционни алгоритми, понеже те генерират себеподобни структури, т.е. сходствата нарастват между микроскопичните и макроскопичните аспекти на една възможна композиция. Тези сходства ясно могат да бъдат видяни в следващия сложен пример, използващ десет различни заместващи правила:

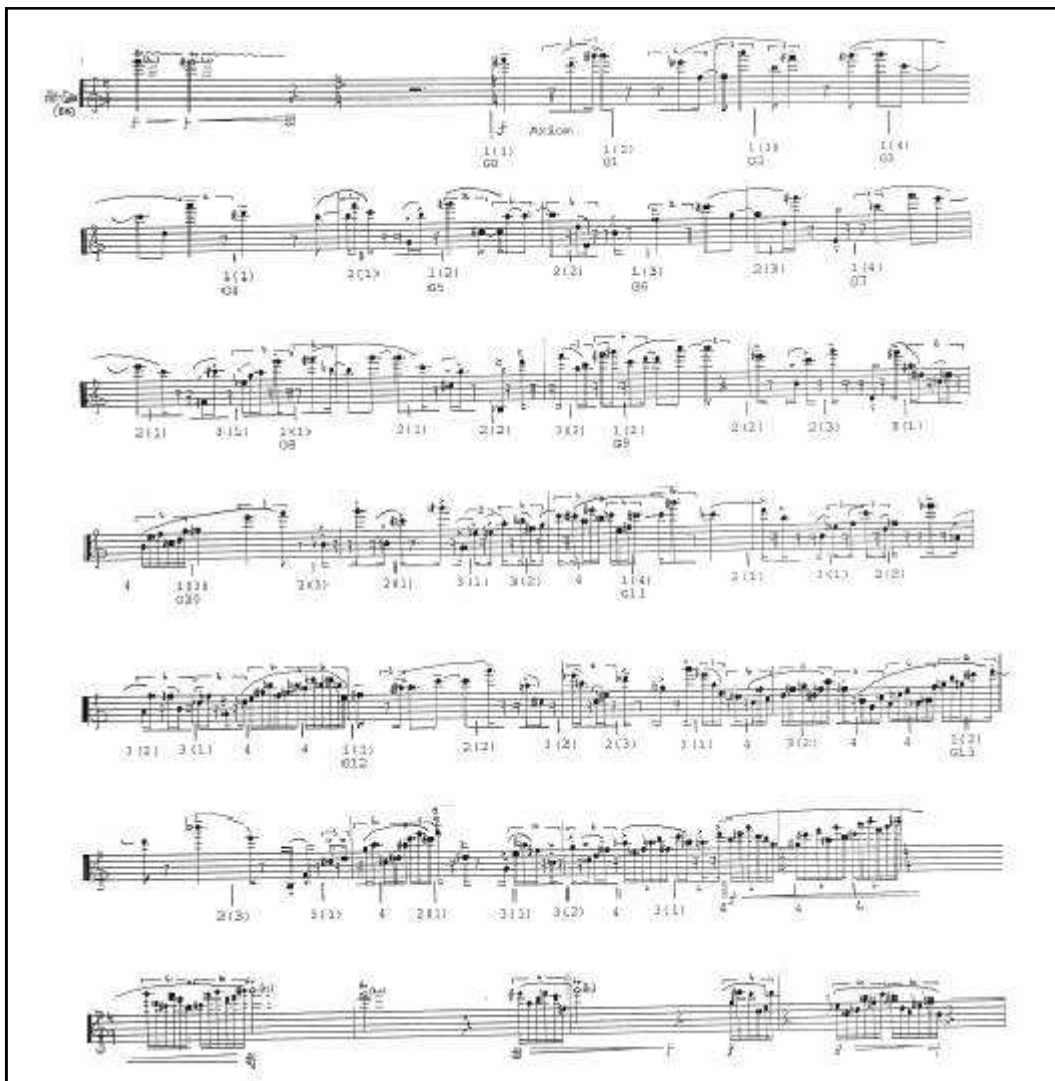
$0 \rightarrow 10$
 $1 \rightarrow 32$
 $2 \rightarrow 3(4)$
 $3 \rightarrow 3$
 $4 \rightarrow 56$
 $5 \rightarrow 37$
 $6 \rightarrow 58$
 $7 \rightarrow 3(9)$
 $8 \rightarrow 50$
 $9 \rightarrow 39$
 $) \rightarrow)$
 $(\rightarrow ($

Ако се започне с аксиомата 4, изниква следната редица от низове:

- 1) 4
- 2) 56

- 3) 3758
- 4) 33(9)3750
- 5) 33(39)33(9)3710
- 6) 33(339)33(39)33(9)3210
- ...
- 10) 33(369)33(359)33(349)33(33(9)3750)33(3758)33(56)33(4)3210

Артистите и музикантите се интересуват от концепцията за себеподобността още от 1980 г. L-системите се използват, за да генерират музикални структури, въпреки че музикалните параметри представени от индивидуални символи в една последователност остават абсолютно свободни. Най-простите опити в тази област са тези, в които символите се използват да представят височината и трайността. [63]



Фигура 17. Пиеза “Клетки” за саксофон и ансамбъл (1993–1994) от Кибуруц. Трети такт, четвърта секция, партията на алт саксофона. [80]

Правилата на аксиома използвани при представянето на музикалните откъси, показани във Фигура 17:

$1(a)$ ако $(a < 4)$, тогава приложи $1(a + 1)$

$1(a)$ ако $(a = 4)$, тогава приложи $1(1) 2(1)$

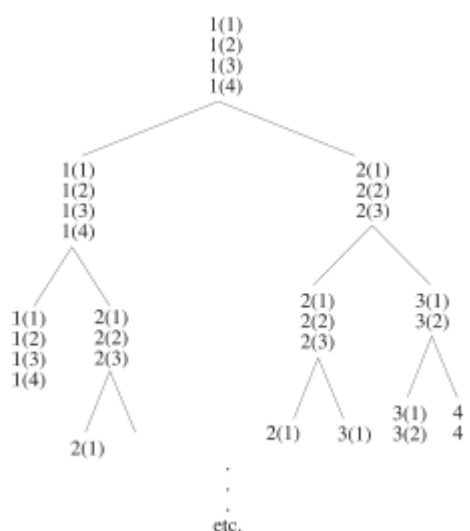
$2(a)$ ако $(a < 3)$, тогава приложи $2(a + 1)$

$2(a)$ ако $(a = 3)$, тогава приложи $2(1) 3(1)$

$3(a)$ ако $(a < 2)$, тогава приложи $3(a + 1)$

$3(a)$ ако $(a = 2)$, тогава приложи $3(1)$

$4\ 4 \rightarrow 4$



Фигура 18. Дървовидна структура нарастваща от правилата показани във фигура 17

Преди да се обърне внимание на теорията на хаоса на конференция през 1989 г., композиторот Ханспетер Кибурц (*Hanspeter Kyburz*) вече е изследвал употребата на L-системите при композиционния процес. Пиесата “*Cells*”, за саксофон и ансамбъл (1993–1994), е един от първите резултати от тези изследвания (Фиг.17) . Кибурц в началото композира малки музикални обекти – малки мотиви, които се съхраняват независимо един от друг в компютъра и могат да се използват като променливи в приложението на L-системите. Правилата на диференциране, конструирани от композитора, определят кога конкретен музикален обект да бъде извикан. Тези правила определят композиционната процедура и не са познати на слушателя, нито могат да бъдат чути. Процесно-ориентираният синтаксис на L-системите предоставя слухово доловимата,

себеподобна структура на композицията. Този метод на композиране е възможно наистина да бъде възприет от слушателя. Когато се използват L-системите, композиторът трябва да определи кога трябва да спре, тъй като процесът става безкраен.

Фигура 17 показва един откъс от партитурата. Демонстрира формалната идея— в този случай себеподобните структури автоматично се определят от употребата на L-системите. Правилата за извеждане показват, че a е променлива, която се замества от числата 1, 2, 3 или 4. Дървото се разделя, ако в $1(a)$ променливата a се замести от 4, или ако в $2(a)$ се разпредели параметър 3. Структурата на дървото произлиза от тези правила показани във Фигура 18. Кибурц използва първите 13 генерации. Те са маркирани в партитурата като G1, G2 . . . G13. Също така се забелязва текущата производна, например 1(2) или 3(1). Тези маркери са свързани директно с по-горе споменатите музикални обекти.

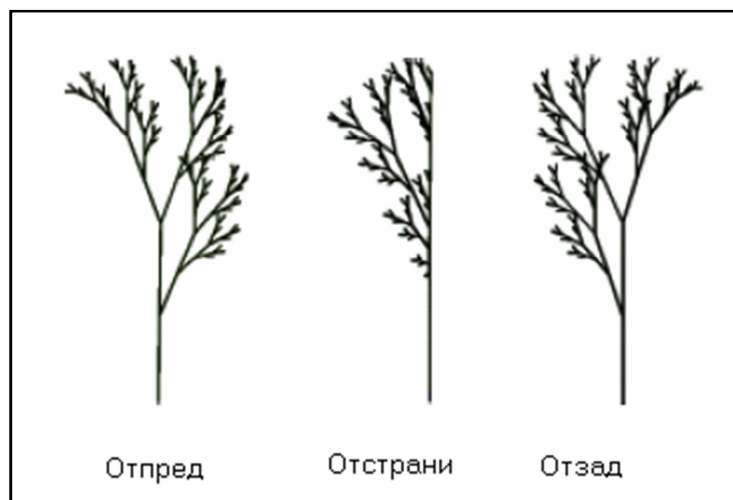
Поради тяхната рекурсивна природа L-системите се изучават от компютърната наука, скоро след като Линденмайер ги предлага. Първото конкретно приложение на L-системите е било в областта на компютърната графика, като помощ в представянето на широката вариантност на фракталите и симулацията на нарастващи растения. Йерархично, стохастично и параметрично L-системите се превръщат в незаменим инструмент за моделиране сложността на растителната морфология, толкова близко, колкото е възможно.

Музикалната интерпретация на L-системите се състои в присъединяване на музикално значение на символите в производния низ (Фиг.19). Програмата LMUSE⁴⁹, интерпретира символите в L-системите като музикални параметри. LMUSE, написан за MS-DOS и JAVA, имплементира както контекстно-свободните, така и контекстно-зависимите, като стохастичните модели на L-системите. LMUSE е триизмерна имплементация на L-системите. В LMUSE състоянието се състои от пространствена позиция, преден вектор, горен вектор и ляв вектор. [80]

Височината се преобразува спрямо предефинирани скали, които могат да бъдат определени от композитора. Трайностите се преобразуват от непрекъснатата скала, което позволява много сложен ритъм. MIDI динамиката също се преобразува от непрекъснатата скала. За да се илюстрира целия процес е изведен пример в таблици 1 и 2.

⁴⁹ LMUSE Ver. 0.7 b, Released 1995, David Sharp;
<http://www.geocities.com/Athens/Academy/8764/lmuse/lmuse.html>; Internet

Графичното представяне на горепосочения стринг е :



Фигура 20. Графично представяне на L-системите [80]

Представени са три различни гледни точки, за да се придаде триизмерната структура. Забележете себеподобността в разклоненията на дървото.(Фиг.20)

Музикалната интерпретация на този низ, съобразно кодово-преобразователната схема се вижда на фигура 21.

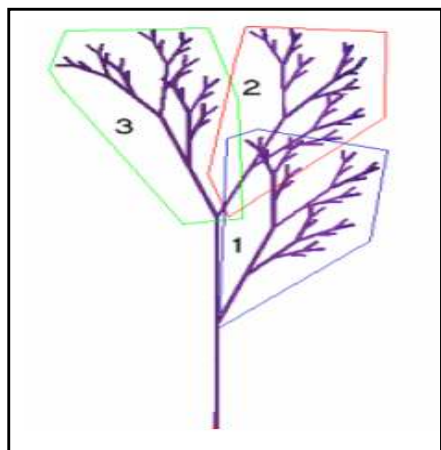
The image displays a musical score for five instruments: Fl. I, Cl., Ob., Fl., and Cello. The score is organized into five systems, each containing five staves. The first system (measures 11-15) shows the initial entries of the instruments. The second system (measures 16-20) continues the musical development. The third system (measures 21-25) shows further instrumental interaction. The fourth system (measures 26-30) and the fifth system (measures 31-36) conclude the excerpt. The notation includes various musical symbols such as notes, rests, and accidentals, indicating a complex polyphonic texture.

Фигура 21. Музика от L-системите [36]

Трябва да отбележим петте нива на разклонение от графичното представяне и как то се интерпретира към петгласна полифония. Първото ниво на разклонение се преобразува към партията на челото, второто към обоя, третото към флейта 2, четвъртото към кларинета и петото към флейта 1. Изборът на инструменти е изцяло в

ръцете на композитора. Ритмичната структура е запазена по-опростена, понеже продължителността е преобразувана към параметър, който е постоянна величина в низа. Последователните еднакви височини са легатирани.

Забележително е как структурата на графичното представяне се запазва и в музикалната интерпретация. За пример, забележете как графичната интерпретация има три основни структури (маркирани като 1, 2 и 3) (Фиг.22):



Фигура 22. Структура на L-системен модел [36]

Тези три структури са всъщност копия на цялото. Този процес на структури в структурите продължава до петото ниво на рекурсия. Трите основни структури кореспондират точно на тактове 1-10, 11- 20 и 21-30 в партитурата. Второто ниво на структурата също е очевидно в музикалната интерпретация. Всяка от тези три секции може да бъде структурирана в три подсекции. Следващият пример (Фиг.23, 24, 25) показва трите под-структури на първата секция (1-10).



Фигура 23. Субструктура 1, тактове 1-3 и първата половина на такт 4 [36]



Фигура 24. Субструктура 2, втора половина на такт 4, тактове 5-6 и първата половина на такт 7 [36]



Фигура 25. Субструктура 3, втора половина на такт 7 и тактова 8-10 [36]

Дори един по-обстоен преглед на тези субструктури показва още по-малки нива на структура, които стигат почти до ниво тонове. Себеподобността на тези структури (както в графичното, така и в музикалното интерпретиране) не е методично точна, а е по-скоро приблизителна. В конкретния пример, това е редица от заместващи правила. В един по-сложен пример, може да бъде достигната статистическа себеподобност на по-високо ниво, чрез стохастични правила. Статистическата себеподобност играе важна роля в музикалната интерпретация на L-системите, защото доставя повече вариации и прави музиката по-малко монотонна и предсказуема. L-системите са едно необятно поле за музикални експерименти. [52] С.Мейсън (*S.Mason*) и М.Сафъл (*M.Saffle*) изследват много тяхното приложение към музикалното композиране.⁵⁰ Много композитори като Гари Лий Нелсън (*Gary Lee Nelson*) в неговата песен “*Summer Song for solo flute*”, си служат с L-системите в техните творби. Манипулирайки правилата и

⁵⁰ S.Mason and M.Saffle, “L-Systems, Melodies and Musical Structure”, *Leonardo Music Journal* 4 (1994): 53-58

преобразуващите схеми, на композитора се предоставя почти неизчерпаем източник за музикална интерпретация и вдъхновение.

4.2.2 КЛЕТЪЧНА АВТОМАТИЗАЦИЯ (CELLULAR AUTOMATA)

Клетъчна автоматизация (*Cellular automata*), за по-кратко ще бъдат срещани в тази глава като СА, са други алгоритми от музикалните дисциплини използвани като метод за алгоритмично композиране. СА са дискретни динамични системи, при които пространството, времето и възможностите могат да имат само определен брой състояния.⁵¹ Те са разработени, за да симулират динамични системи, подобно на движенията на течностите. [59] Например, в течността енергията и една частица течност ще бъдат представени от клетка. Индивидуалните частици си влияят една на друга като се блъскат една в друга или променят местата си, като по този начин се определя движението на течността като цяло. Най-разбираемите, елементарни ефекти в пространството на една клетка незабавно могат да предизвикат неочакван ефект в системата като цяло, защото всяка клетка може едновременно да влияе и да бъде повлияна от няколко от нейните съседни клетки. Идеите, които стоят зад СА са окуражили различни музиканти да ги използват като модели за композиционните процеси.⁵² За Ксенакис тези идеи са инструмент конструиращи сложни структури с минимални средства [14]. Процесът на разработка на СА определя динамичното развитие на оркестралните клъстери в “*Horos*” (1986). Приложението на музикални алгоритми, симулиращи естествен феномен, повдига въпроса дали композиторите тайно възприемат алгоритмичното композиране като начин за естествено генериране на естествени форми — форми, които доказват самите себе си чрез собствената си пълнота от естественост. Във всеки случай, моделирането на естествени процеси изисква употребата на компютрите, на които се дължат огромния брой математически изчисления, необходими при моделирането на такива процеси. Компютърът не се използва само като композиционно удобство—той се превръща в неизбежно средство в този композиционен процес.

⁵¹ Miranda, “Music Composition Using Cellular Automata”, 105-117

⁵² Бейлс отбелязва, че “Като композитор аз се интересувам от еволюционните модели, които нарастват в по-голяма степен от теориите за структурен дизайн”

СА е много богат източник за алгоритмично композиране. [33] Композиционното приложение на СА се открива в работата на Хънт, Кърк, Ортън, Милън и Миранда.⁵³ Теорията за СА бива представена от Джон фон Нойман (*John von Neuman*) през 1960 г., като модел за биологично себе-репродуциране. СА има две основни характеристики:

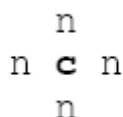
1. Правилна, симетрична n -мерна решетка, където всяка клетка в тази решетка има отделно състояние в даден момент.

2. Динамично поведение, контролирано от редица правила. Тези правила определят състоянието на клетките през генерираните стъпки, в зависимост от състоянието на съседните клетки.

Клетките в СА са като устройства за памет, които съхраняват автоматизираните състояния. Най-простият случай включва само две възможни състояния за всяка клетка – 0 (несъществуващ), 1 (съществуващ). В някои по-сложни СА, клетките могат да имат много повече състояния. Клетките са подредени в n -мерна решетка. При едномерна решетка на СА, клетките са подредени на една линия. При двумерна СА клетките са подредени върху плоска решетка (двумерна матрица). Триизмерната СА е подредена в кубична матрица.

Правилата, които контролират динамичното поведение на СА се отразяват на всяка клетка в решетката. Тези правила определят състоянието на всяка клетка в последващите генерации. Трябва да бъдат взети под внимание три основни типа съседни клетки, когато се дефинират правилата (като в случая щ обърнем внимание на двумерната СА):

1. Така нареченото фон Нойман (*von Neumann*) съседство, където всяка клетка има четири съседа (Север, Юг, Изток, Запад), като радиуса на това съседство е 1, тъй като само близкостоящите клетки са взети под внимание:



2. Мур (*Moore*) съседство, където всяка клетка има 8 съседа (N, NE, NW, S, SE, SW, E, W клетки). Радиусът и тук е 1 :

⁵³ A.Hunt, R. Kirk, and R. Orton, "Musical Applications of a Cellular Automata Workstation", in Proceedings of the 1991 International Computer Music Conference (San Francisco: International Computer Music Association, 1991), 165-168

```

n n n
n c n
n n n

```

3. Разширеното съседство на Мур, където са взети под внимание клетките отвъд радиуса на съседство. Радиусът може да бъде 2 или повече от това:

```

n n n n n
n n n n n
n n c n n
n n n n n
n n n n n

```

За едномерна СА, съседствата фон Нойман и Мур са идентични:

```

n c n

```

Разширеното съседство на Мур за едномерна СА ще бъде:

```

n n c n n

```

Правилата, които определят динамичното поведение на системата могат да бъдат категоризирани в два класа:

1. Състоянието на дадена клетка се определя чрез индивидуално изследване поведението на всички съседни клетки .
2. Неограничени правила, където поведението на която и да е клетка се определя от съвкупното състояние на съседните ѝ клетки.

Поведението на СА може да бъде класифицирано в четири основни класа:

Клас 1 – ограничена точка. След определен брой генерации се достига стабилно положение от системата.

Клас 2 – ограничен цикъл. Този тип се развива към стабилно състояние, където образците се повтарят периодично.

Клас 3 - системата генерира апериодични, хаотични образци от близкостоящите условия. Образците могат да наподобяват себеподобността на фракталната крива.

Клас 4 - поведението на системата е сложно, но не и хаотично. Този клас е единствения, който е способен да изпълнява универсално изчисление, което означава, че е способно да продължава който и да е определен алгоритъм.

4.2.1.1 ЕДНОМЕРНА СА

Ще разгледаме едномерна СА, предложена през 1982 г. от С. Уолфрам (*S. Wolfram*). [77] Този тип автоматизация, поради своята простота, е изследвана значително. На всяка клетка са позволени две състояния (0 – не съществува, 1 – съществува). Съседството от типа на Мур – всяка клетка има два съседа, отляво и отдясно. Този тип СА позволява 256 възможни правила: на всички три клетки в съседство са позволени две състояния, което дава 8 (2^3) възможни комбинации на всички клетки и всяка една от тях може да избере измежду тези две състояния в следващата генерация. Възможния брой правила поради тази причина е $2^8 = 256$.

Възможни конфигурации на клетка:

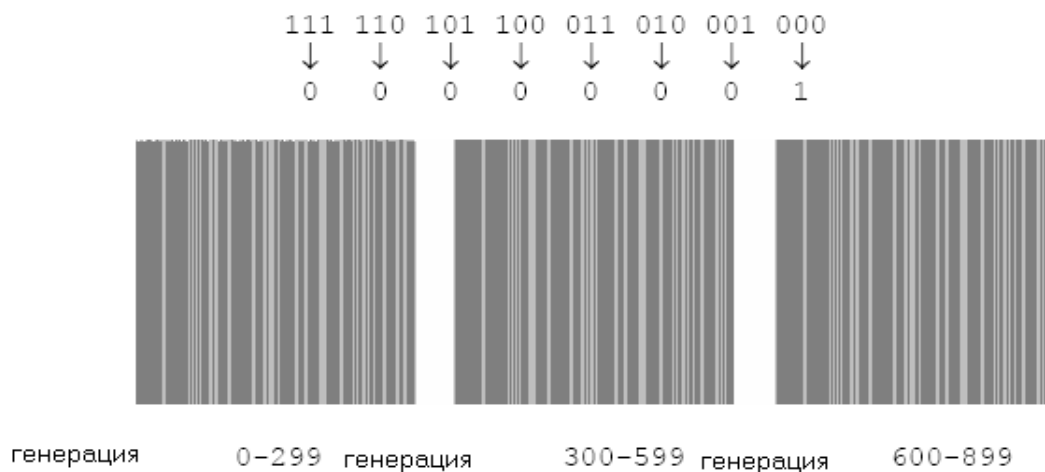
111	110	101	100	011	010	001	000
↓	↓	↓	↓	↓	↓	↓	↓
1	1	1	1	1	1	1	1
или							
0	0	0	0	0	0	0	0

На всяко правило съответства 8-битов низ, който можем да разглеждаме като двоичното представяне на номера на правилото. Например правило 30 в двоичното му представяне е 00011110 и тогава :

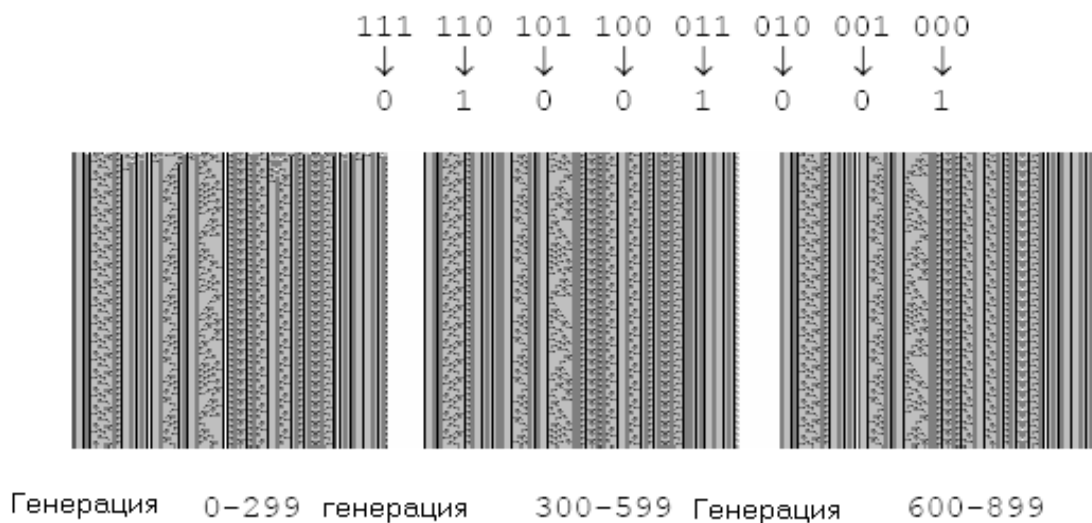
111	110	101	100	011	010	001	000
↓	↓	↓	↓	↓	↓	↓	↓
0	0	0	1	1	1	1	0

Правилата определят състоянието на всяка клетка в следващата генерация в зависимост от собственото ѝ състояние и това на съседите ѝ в текущата генерация. Едномерната автоматизация се разгръща върху една редица от клетки. Генерациите или времевите стъпки са представени една след друга, подобно на разгърнато одеяло. Всяка

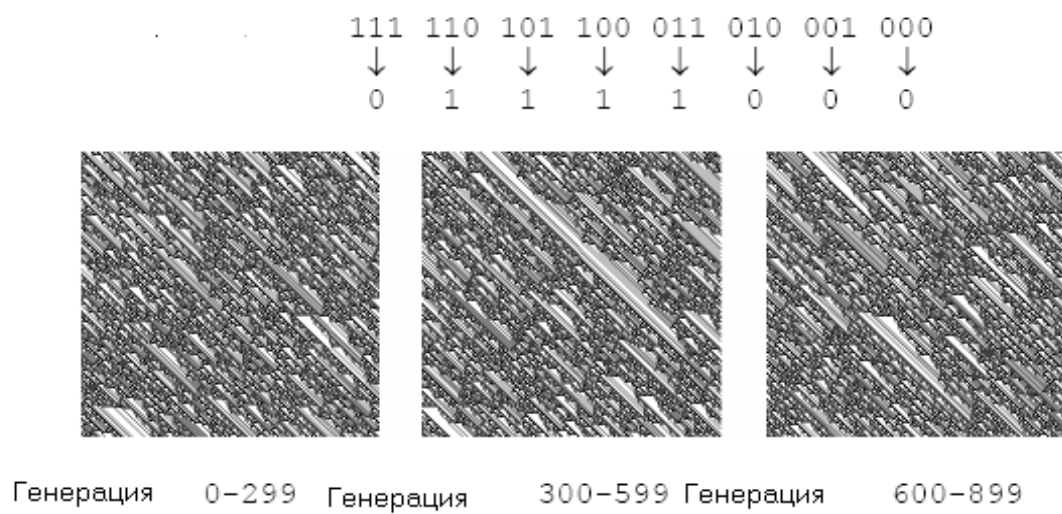
нишка от това одеяло представя една времева стъпка или генерация на клетъчната автоматизация. Това което следва на фигурите 26, 27, 28 и 29 по-долу, показва динамичното развитие на тази едномерна автоматизация при четири различни правила. За всяка система са изчислени 900 генерации. Решетката е широка 300 клетки. Клетките, които се намират на границите на решетката, са съседни на всички останали.



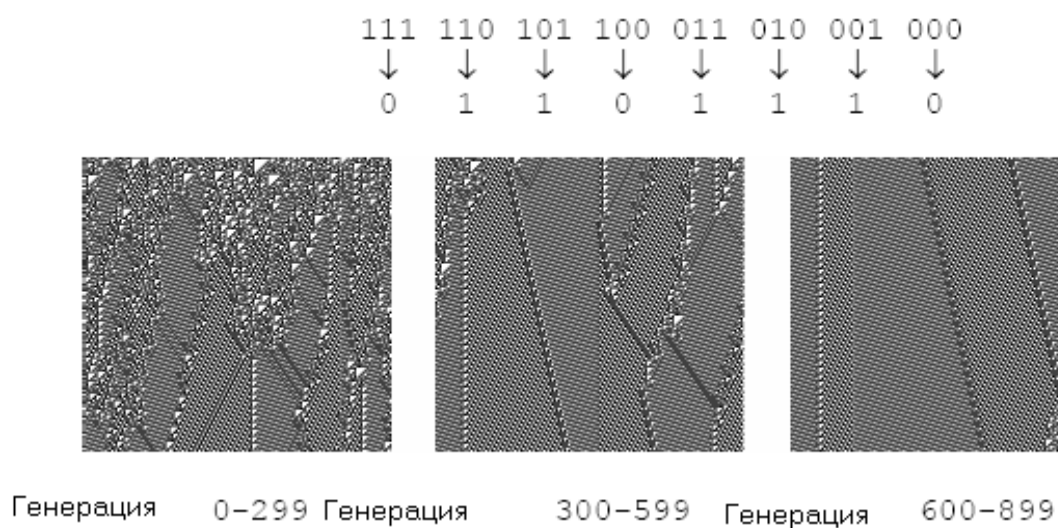
Фигура 26. Правило 1, Клас 1 [134]



Фигура 27. Правило 73, Клас 2 [134]



Фигура 28. Правило 120, Клас 3 [134]



Фигура 29. Правило 110, Клас 4 [134]

За генерация 0, състоянията на клетките са инициализирани случайно.

Преобразувайки едномерната клетъчна автоматизация към музика включва транскрибиране на автоматизационната еволюция към музикално събитийно пространство. Всяка генерация е преобразувана към членовете на избраното от събитийно музикално пространство. Докато всяка генерация включва колекция от клетки, трябва да бъде направена директна връзка между всяка възможна

конфигурация на клетките в решетката и събитийното пространство. За да се постигне това, трябва да се кодира всяка възможна комбинация на клетките в решетката с уникално число. В примера s е броя възможни състояния за една клетка, n е броя на клетките в решетката (подредени от 0 до $n-1$) и t_p е състоянието на една клетка в позиция p ($p \in \{0, n-1\}$). Всяка възможна конфигурация на клетки в генерацията е кодирана с уникално число, както следва:

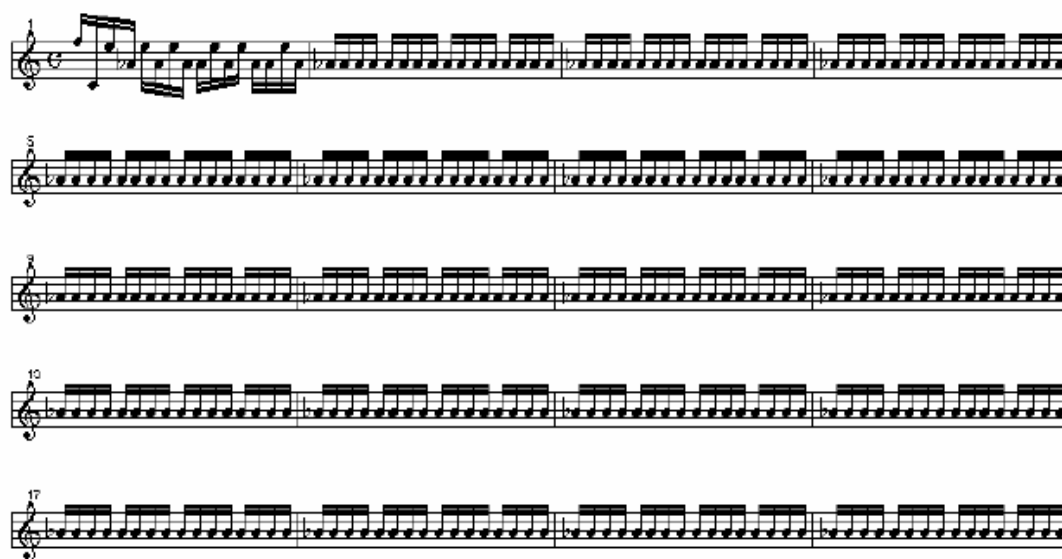
$$u = t_0s^0 + t_1s^1 + t_2s^2 + \dots + t_{n-1}s^{n-1}$$

След това числото се преобразува към музикалното пространство чрез модулна операция:

$$\text{Номер на събитие} = u \bmod (\text{броя елементи в събитийното пространство})$$

Следвайки тази схема, четири конфигурации кореспондиращи на правила 172 – Клас 1, 73 – Клас 2, 30 – Клас 3 и 54 – Клас 4, се преобразуват към музикалното събитийно пространство, което се състои от хроматична гама в две октави от C_4 до B_5 . Решетката има 64 клетки. Следователно броя възможни конфигурации на клетките е 2^{64} .

Правилата са избрани да покажат как характеристиките на различните класове са транскрибирани към музика. От всяка конфигурация са преобразувани 3020 генерации (при тези примери, нотните стойности са избрани произволно). Музикалният превод е както следва на фигури 30, 31, 32 и 33:



Фигура 30. Музика от правило 172, Клас 1 [95]



Фигура 31. Музика от правило 73, Клас 2 [95]



Фигура 32. Музика от правило 30, Клас 3 [95]



Фигура 33. Музика от правило 54, Клас 4 [95]

Подобието на тези музикални последования към характеристиките на кореспондиращия автоматизиран клас е открояващо се. Забележете как на фигура 30 примерът бързо се стабилизира към уникален член на събитийното пространство. Това е характеристика на Клас 1, който има едноточкови атрактори.⁵⁴ Примерът от фигура 31 показва как ограничената характеристика на Клас 2 е преведена в музика. Забележете повтарящия се 60-тонов цикъл, започващ от третото време на 4 такт.

Хаотичното поведение на Клас 3 се демонстрира в примера от фигура 32. Не могат да бъдат открити забележими образци. Наподобява на случайно избрана колекция от височини, въпреки че процедурата е напълно детерминирана. Примерът от фигура 33 е най-приятния откъм музикална гледна точка. Този пример показва поведението на автоматизацията на Клас 4. Музиката е в пряка връзка с автоматизационния капацитет на информационното разпространение. Няма я предсказуемостта на примерите от фигури 30 и 31, както и произволния характер на пример 32. Всеки тон или събитие в примерните музикални последования представя една генерация в развитието на автоматизацията, както и дефинирана конфигурация на клетки. Тази кодово-преобразуваща схема запазва характеристиките на автоматизацията, когато се преобразува към музикално събитийно пространство.

⁵⁴ Виж речника

Увеличавайки броя състояния за една клетка, възможната редица от правила нараства експоненциално и поради това типа на клетъчна автоматизация, която може да бъде генерирана, позволява 3 състояния за клетка вместо две, увеличавайки броя възможни правила от 256 на 7,625,597,484,987 (3^{27}).

Двумерната клетъчна автоматизация се разгръща в двумерна матрица или мрежа от клетки. Поведението на двумерната клетъчна автоматизация е далеч по-сложно и предлага много повече възможности за експерименти. Управлява се от редица от правила, които определят поведението на системата. [77]

4.2.3 ФРАКТАЛИ

Тъй като една от темите на изследването е фракталната музика ще обърнем внимание върху понятието фрактал, като математическо понятие, използвано във всички сфери на изкуството.⁵⁵ Фракталите и фракталната геометрия се раждат в опит да се усъвършенства математическата рамка, за да се разбере начина, по който природата използва едни и същи форми в одушевените и неодушевените неща [50]. Традиционната геометрия винаги е боравила с правилни форми и криви. За разлика от нея, формите в природата не могат да бъдат описани правдоподобно с класическата геометрия.⁵⁶

Една основна характеристика на фракталите е тяхната себеподобност на всички нива: една малка част от целия обект изглежда като цялото. [19]

Себеподобността се изразява по два начина: точен, в който увеличени малки части от обекта са идентични на цялото; статистически, в който една увеличена част от обекта има същите статистически възможности като цялото. Някои фрактални обекти, като известната редица на Манделброт, не са себеподобни.⁵⁷

Фракталите са както естествени, така и математически обекти. Фракталите в природата, например, разклоненията на едно дърво имат фрактална структура: индивидуалните клони изглеждат като умалени версия на цялото дърво.

⁵⁵ Думата фрактал е въведена от полския математик Беноа Манделброт, от латинската дума *fractus*, означаваща счупен, надробен, фрагментиран.

⁵⁶ Както казва и самия Манделброт, “облаците не са сфери, планините не са конуси, крайбрежните ивици не са кръгове и кората на дървото не е гладка.”

⁵⁷ Briggs, *Turbulent Mirror*, 90

Себеподобността в естествените фрактали винаги е статистическа: няма точна себеподобност в природата.

По-долу са дадени примери, които доказват, че фракталната структура е залегнала в музикалните и графични изображения. Математическите фрактали, които са единствените подходящи за алгоритмично композиране, са повлияни от нелинейните уравнения. Някой ще очаква дори, че генерирането на толкова сложни форми ще изисква също така и сложни уравнения, но в действителност това не е така. От откриването на фракталите и науката за хаоса, са открити и предложени хиляди фрактални уравнения. От всички тях, ще спрем вниманието си на първия и най-известен – фракталът на Манделброт. Той се дефинира посредством така нареченото *множество на Манделброт* от комплексни числа. Да разгледаме рекурентно зададената редица

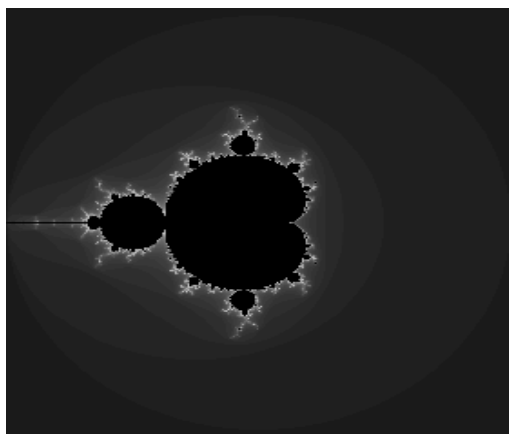
$$z_0 = 0, \quad z_{n+1} = z_n^2 + c,$$

където c е комплексно число, явяващо се параметър. Ако тази редица е ограничена, то тогава казваме, че числото c е елемент на множеството на Манделброт. С други думи, множеството на Манделброт M се дефинира формално като

$$M = \{c \in \mathbb{C} : \exists t \in \mathbb{R}, \forall n \in \mathbb{N}, |z_n| \leq t\}.$$

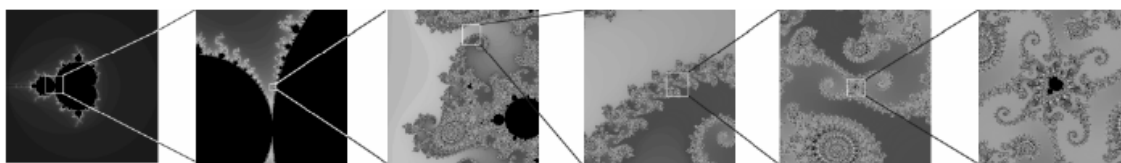
Оказва се, че множеството на Манделброт е компактно в комплексната равнина, като е ограничено от кръг с радиус 2 и център $(0,0)$, т.е. достатъчно е да вземем $t = 2$.

За да се създаде един фрактален образ, на всяко комплексно число c съпоставяме точка от комплексната равнина и я оцветяваме в черно, ако $c \in M$ или съобразно минималния индекс $n \in \mathbb{N}$, за който $|z_n| > 2$. За да може този процес да е краен е необходимо да дефинираме максимално допустима стойност за броят итерации m , т.е. процесът спира, когато $|z_n| > 2$ или $n > m$. Ето как изглежда една реализация на горната процедура. (Фиг.34):



Фигура 34. Фракталът на Манделброт [109]

Тъй като множеството на комплексните числа е гъсто, то можем да представим математически „увеличението” на фрактала, показвайки неговата сложност и безкрайно детайлна структура. (Фиг.35).



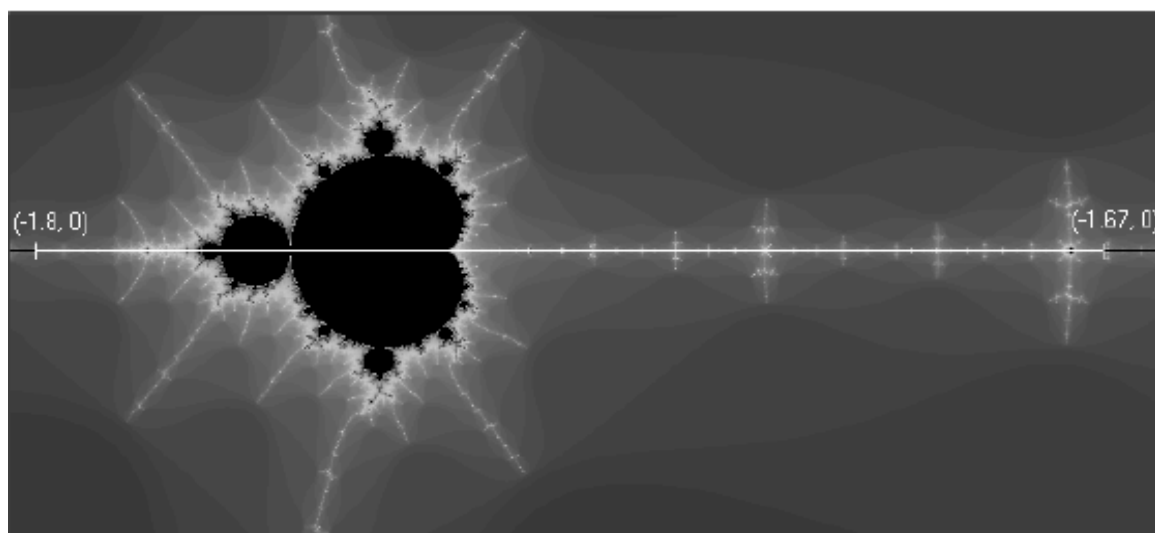
Фигура 35. Шест увеличения на фрактала на Манделброт [109]

Един метод да се преобразуват фрактали е да се изчислят итерационните стойности на редицата от точки в едномерен дял в двуизмерната структура на фрактала и да се преобразуват тези стойности към музикално събитийно пространство. Това се изпълнява по следния начин:

1. Взимат се които и да са две точки във фракталния образ, една начална точка (x , y) и една крайна точка (Δx , Δy).
2. Изчисляват се стойностите на итерацията за точки, които лежат в сегмента от (x, y) до $(\Delta x, \Delta y)$, линейно.
3. Преобразуват се итерационните стойности на този ред от точки към музикално събитийно пространство.

Като пример, ще разгледаме кодово-графично преобразуване на един дял от редицата на Манделброт. Събитийното пространство, за улеснение е 4 октавова хроматична гама (от C_2 до B_5 – 48 тона, подредени от 0 до 47), както и тишина за точките, които не избягват безкрайността (черните региони от редицата). Избират се

шестнадесетините като ритмични стойности. В допълнение, ако две последователни точки имат една и съща итерационна стойност, техните преобразуващи височини ще бъдат легатирани. Итерационния лимит е настроен до 1000. Сегментът ще бъде разделен на 600 точки. Визуално, се преобразува следния сегмент от редицата на Манделброт (фиг.36):



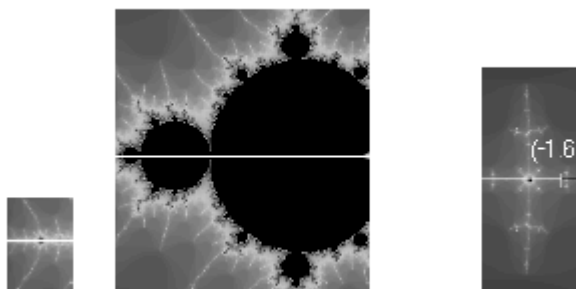
Фигура 36. Преобразуваща секция от редицата на Манделброт [109]

Преобразуваният сегмент преминава от $(-1.8, 0)$ до $(-1.67, 0)$. Итерационните стойности се преобразуват към събитийното пространство от 48 елемента, използвайки нормализиран преобразуващ метод. Това подsigурява, че поведението на итерационните стойности правдоподобно се възпроизвежда към събитийното пространство, което е избрано. Генерираното музикално последование е както е показано на фигура 37:



Фигура 37. Музика от редицата на Манделброт [125]

Сравнявайки сегмента от редицата на Манделброт и неговата музикална транскрипция, може наистина да се види как чертите от преобразувания откъс са реализирани музикално. За пример, ще обърнем внимание на откъса тишина в тактове 4-5, 6-15 и 37. Те кореспондират на откъсите върху сегмента от точки, които са от фрактала на Манделброт. Тези откъси от сегмента са (виж фиг.38):



Фигура 38. Откъси от сегмента [109]

Забелязваме, че през повечето време музиката се намира в по-долен регистър, понеже итерационните стойности на тези точки остават ниски.

Тази техника може да бъде разширена чрез кодово-графично преобразуване на голям брой близкостоящи сегменти и наслаждайки ги полифонично. Това всъщност ефективно ще представи един вид „сканиране” на повърхността на фрактала. Този метод несъмнено ще преобразува по-богато структурата на целия фрактал отколкото един едномерен дял. Разбира се, не сме ограничени само във височина, понеже същия преобразуващ метод може да бъде приложен едновременно към трайност, динамика, тембър. [50]

4.2.4 МЕТОДИ ЗА СЪЗДАВАНЕ НА ФРАКТАЛНА МУЗИКА

Ако съпоставим един фрактал в чистата му геометрична форма с музиката и създаването на фрактална музика, всъщност ще достигнем до извода, че той представлява самата основа – формулата, под формата на един безкраен, повтарящ се алгоритъм, на базата на който се създава музика, което пък сравнено с класическите похвати за композиране – е идентично с процеса на композиране в традиционната музика. В музиката и в нейните основни композиционни техники винаги има заложен замисъл или т.нар. *Лайт мотив*, който без значение колко е сложен и какво съдържание носи в себе си – подлежи на множество разработки и надграждания, в които вглеждайки се или по-точно вслушвайки се – откриваме малкия, първоначален мотив, станал основа за създаването на цяла симфония. (*Приложение 3, Bethoven - 5th Symphony*)

Самата композиционна техника може да не е нищо особено, подобно и на един алгоритъм и затова мелодическата линия е равностойна като значение при създаването на един интересен алгоритъм.

Не е достатъчно да имаме форма. Нужно е тя да бъде изпълнена със съдържание и то да отговаря на вижданията и търсенията на самия композитор, което е един от основополагащите закони при принципите за композиране. Именно чрез един интересен алгоритъм изпълваме със съдържание музиката, която се създава, като го допълваме със звуковите характеристики, към които се стремим.

Компютрите композират фрактална музика използвайки винаги формални алгоритми - софтуерни програми, за да създават музика.

Една музикална идея трябва да бъде преведена и трансформирана в съдържание, като разликата между традиционните принципи на композиране и тези, които използват за композиране определен софтуер, е в самата идея, която цели да внуши композитора.

При традиционната музика композиторът сам избира правилата за композиране и съответно прозвучаване на музиката му – той избира мелодията, хармонията, изразните средства.

Докато при алгоритмичната музика композиторът борави със заложените неизменни алгоритми. Но в този случай отново има избор, защото може да избере самия алгоритъм, който да отговаря на търсенията му, както и може да зададе основни параметри, които този алгоритъм да използва при генерирането на музика.

Тежестта на избора на всеки композитор, който използва софтуер за композиране на алгоритмична музика, се състои в това, че вследствие избора на определен лад, тоналност, тембър, вариации, на базата на всеки един избор, се изгражда и самата алгоритмична композиция. Ако тя не бъде възприета и харесана от публиката, композиторът няма да може да обвини само алгоритъма, който е използвал, защото той задава самата форма, но съдържанието бива запълнено и изпълнено с усета и търсенията на самия композитор.

Алгоритмичното композиране е техниката, която използва алгоритми за създаване на музика. Алгоритмите са били използвани от столетия за композиране на музика. Похватът, който се използва да се изчертае водещия глас в западния контрапункт, може да бъде сведен и до алгоритмичната детерминанта. Терминът обикновено се използва, за да обясни употребата на формални процедури за създаването на музика без човешка намеса, използвайки случайни процедури или компютри.

Много от алгоритмите, които нямат непосредствена музикална връзка се използват от композиторите, като творческо вдъхновение за тяхната музика. Успехът или провала на такива процедури като източници на добра музика, зависят до голяма степен от преобразуващата система, използвана от композитора да транслира немусикалната информация в музикален поток от данни.

Повечето програми, които генерират алгоритмична музика могат да бъдат намерени и изтеглени от Интернет. Една от най-важните черти на тази програми е че създават MIDI файлове от алгоритмичните процеси, които представят. Тези файлове

по-късно могат да бъдат използвани в други секвенсерни програми, където самите файлове да бъдат редактирани допълнително. Тези файлове могат да бъдат заредени в музикално нотиращи или секвенсерни програми, за по-нататъшна обработка и редактиране.

Алгоритмичната техника за създаване на музика е толкова многостранна, че можете да я използвате, за да създавате различен стил музика, както и да добавяте нови елементи към музиката си.



Фигура 39. Фракталните образи се генерират от прости математически формули, които са много богати в детайлите. Това се дължи на употребата на итерации и лупове [116]

Конструкцията на подобни сложни форми предполага комплексни правила, но всъщност алгоритмите или уравненията, които генерират фракталите са сравнително прости. Техният визуален резултат, показва голямо великолепие (Фиг.39).

4.2.5 МУЗИКАТА НА ФРАКТАЛИТЕ

Експериментите, чрез които учените се опитват да покажат красотата на фракталното изкуство, разширено и в електроакустичната музика, понякога дават интересни резултати, но често звучат твърде странно. Десетичната система, която се базира на числата от 0 до 9 не е единствения възможен метод да се представят

естествените числа. Най-простото представяне е в двоична бройна система, която използва само цифрите 0 и 1. Компютрите винаги изчисляват процесите, използвайки двоичната бройна система, защото е най-лесния начин за комуникация с електрическите устройства: наличието на ток означава 1, а липсата му се означава с 0. [24]

4.2.5.1 ФРАКТАЛИ И ХАОС

Термините фрактали и хаос често се споменават заедно, но по същество са напълно различни. Фракталите са обекти. Хаосът може да се определи като непредсказуемо показване на поведение или с други думи, като детерминиран процес, който е много чувствителен при първоначалните условия. Хаотичните процеси могат да представят много разнообразни и различни резултати, при много малки разлики в стартовите условия. Въпреки че фракталите са сложни 'безкрайно детайлни' обекти, те са забележително лесни за създаване, особено със средствата на компютъра.

4.2.5.2 СПЕЦИФИКА НА ФРАКТАЛНАТА МУЗИКА

Много от атрактор-базираните алгоритмични композиции работят като задават начална настройка на отделните x , y координати върху атрактора. След това те проследяват орбитата на успешните итерации върху атрактора и преобразуват, както абсолютните стойности на координатите, така и промените в координатите за даден период от време. Например, при една графика хоризонталната ос представя времето, а вертикалната представя нотата или броя музикални събития. При това числово/музикално кодово-графично преобразуване, обикновени мелодии (или колекции от музикални събития) са подобни на асиметрични функции.

Фрактализацията е свързана с информационното съдържание и детайлизиране в една система – природна, геометрична или музикална.

Фракталите съществуват между хаоса и реда – това важи и за фракталната музика.

Типичен пример за фрактална музика е т.нар. алгоритъм *Logistic map* :

$$x_{n+1} = kx_n(1 - x_n),$$

където k е реален параметър в интервала $[0, 4]$. Итерацията на тази формула връща като резултат стойности между $[0, 1]$. Основата на тази редица зависи от параметъра k . Ако $k \leq 3$, то редицата клони към фиксирана стойност. Докато, ако

$k > 3$ и $k < 4$, то редицата няма да е сходяща. Поведението на този алгоритъм е хаотично, което всъщност е основата на фракталната природа.

Такива рекурентни редици са основата за създаване на фрактална музика, като елементите им се преобразуват към тонове от музикалната система.

Друг пример за такъв тип алгоритъм е *Henon map* :

$$x_{n+1} = 1 + y_n - ax_n^2,$$

$$y_{n+1} = bx_n,$$

в който a и b са реални параметри. По подразбиране $a = 1.4, b = 0.3$. При нотирането с *Logistic map* се използва принципен метод на фрактален модул, за да се преобразуват числата към поредица от тонове, докато нотирането с *Henon map* се конкретизира в самия скрипт.

Разбира се, че може да се създават хиляди такива хаотични поредици от тонове – на това ниво във фракталната музика липсват компоненти, като динамика, темпо, трайност, тембър. Това вече може да бъде добавено към даден скрипт или да се използва конкретно приложение за създаване на музика.

4.2.6 ПРИЛОЖЕНИЕ НА ФРАКТАЛНОСТТА В МУЗИКАТА

В музиката приложението на фракталната геометрия е в три основни области: композиция, синтез на звука, аналитични изследвания. Според класификацията на холандския музиковед Н. Путеманс, която разгледах по-рано в четвърта глава, има 4 основни начина за използването на фрактали в композиция [23] :

1. Конвертиране на графично изображение - изображението на фрактала се конвертира в двумерна партитура (координати височина-време), като се отчита графичното разположение на точките и техния цвят;
2. По формула - определяне на височините и продължителността на звуците чрез аритметични пресмятания съгласно фрактални алгоритми;
3. По генезис - определен музикален параметър се трансформира така, че резултатът от това запълване демонстрира фрактален характер;
4. Фрактална композиция чрез игрово поведение - алеаторични структури, в които изпълнителите си вазимоделират на базата на някакви правила, създавайки в пиесата

модел на поведение в сложни системи. Повечето композитори предпочитат една или две категории от изброените. Главно това са Ч. Додж, Г. Нелсън, М. МакНаб, Б. Еванс, Л. Остин, Ч. Вуоринен, Лигети и Литъл. (*Приложение 3, György Ligeti (1923-2006) - Volumina (1961 62) part one; György Ligeti Lux Aeterna*)

Друго виждане за фракталната музика е чрез опит да се генерира музикален материал чрез смисила на себеподобността. Фракталоподобни форми могат да бъдат генериране чрез прилагане на итерационно правило върху даден мотив. Този тип генеративен процес в математиката има наименования като *translation*, *reflection*, *rotation* - или трансляция на основни геометрични форми със симетрично-групирани операции. Аналогични операции на тези могат да бъдат открити и в музиката: транспониране, инверсия, увеличаване и умалчаване. Тези трансформации върху първоначални мотиви могат да се открият в повечето контрапунктична музика, като канони и фуги. Всъщност, от Ренесанса до времето на Бах, композиторите са създавали канони-гатанки или пъзели канони, в които една проста музикална фигура е била давана като основа на изпълнителя, който е трябвало да бъде запознат с правилата на канона, за да може да развие тази основа по свое осматрление. Трябвало е да решават интервалите за имитациите и дали и кога да се прилагат различни симетрични операции. Бах от своя страна е имал невероятна дарба за импровизиране върху такива откъси. Като пример, канонът даден по-долу, принадлежи на Бах и е пъзел (Фиг.40). Предложен е като основа за създаването на канон от осем гласа, като съдържа само височинна и метрична информация. Следващата фигура представя решение за пъзела, предоставено от композитора Лари Соломон (*Larry Solomon*), където времеви интервал за влизането на гласовете е четвъртина нота (Фиг.41). Височинния интервал за имитацията е чиста квинта, връщайки се обратно към тониката, след което имитацията се обръща и транспонира към редуващи се встъпления.

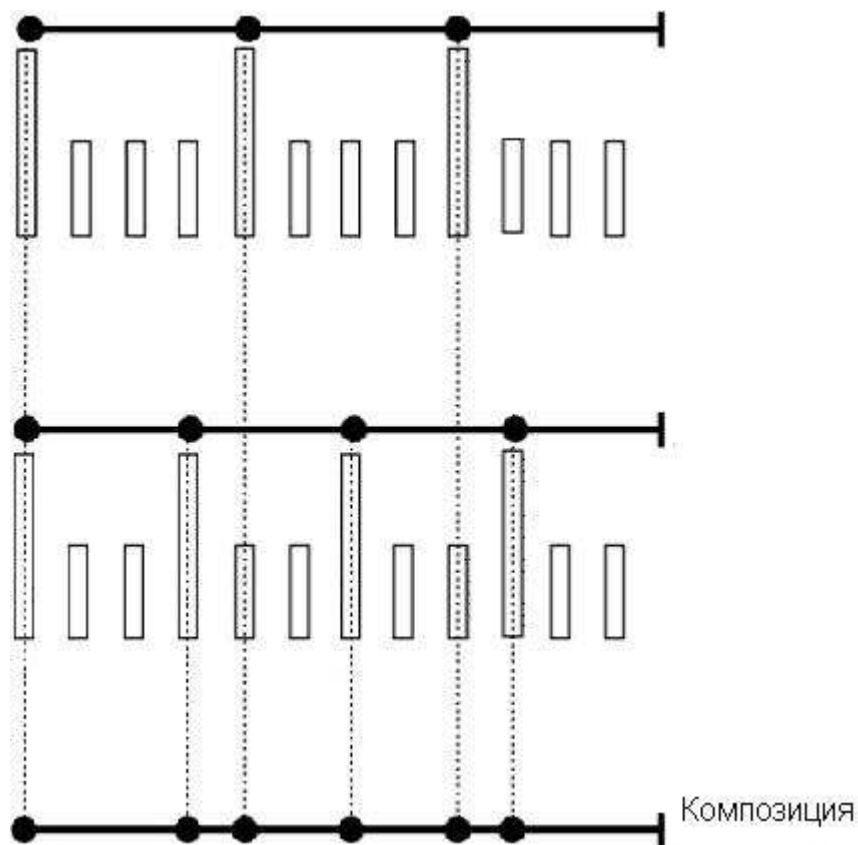


Фигура 40. Идея за канон-пъзел [135]



Фигура 41. Едно възможно решение на пъзелния канон [135]

Ритъмът също така несъмнено е интересно пространство, за композиране на фракталоподобни структури. През 1941 Джоузеф Шилингер (*Joseph Shillinger*) публикува *Shillinger System of Musical Composition*, 12 тома работа, в която се предсказват много от откритията свързани с алгоритмичното композиране, години преди да се създаде компютъра. [66] В първия том, Шилингер се занимава основно с теорията на ритъма. Шилингер е имал идеята, че сложните ритми могат да бъдат формирани чрез наслагване на два различни потока и формирайки нов поток, който комбинира подходящото начало от двата потока. Като пример нека вземем под внимание поток на период три и поток на период 4, показани по-долу (Фиг.42). Ако приемем 12 като най-ниския общ знаменател за тези два потока, то първият ще се състои от три тона, с размер $4/4$, докато вторият ще се състои от четири тона с размер $3/4$. След това нека обединим тези два потока, като получения поток ще се състои от тоновете и от двата потока.



Фигура 42. Ритмичната система на Шилингер [129]

4.2.6.1 ЛИНЕЙНА ТРАНСФОРМАЦИЯ (*THE LINEAR TRANSFORMATION*)

Барнсли обяснява, че линейната трансформация предоставя една нова двойка числа за въвеждане на хоризонтални и вертикални координати. Това може да бъде идея и за разширяване или сливане на цялото числово пространство.

Музикантите и композиторите работят с тези трансформации интуитивно, но се отнасят към тях с различна понятийна система от тази на математиците. По-долу се описва тази система.

TRANSPOSITION - Транспонирането, което означава преместване между две позиции от оста, в музиката означава изсвирване на тона по-висока или по-ниска височина от оригиналния.

$$\text{new note} = \text{note} + \text{constant}, \quad \text{new position} = \text{position}$$

CHROMATIC INVERSION - Хроматичната инверсия е подобна на отражение около дефинираната ос. Изсвирва се наопаки.

$$\text{new note} = -\text{note} + (2 * \text{axis}), \quad \text{new position} = \text{position}$$

RETROGRADE - Обърнато наопаки е когато се свири музикалното последование в обратен ред, което може да се приеме като отражение + нормализираща трансформация

$$\text{new note} = \text{note}, \quad \text{new position} = -\text{position} + \text{endpoint}$$

AUGMENTATION - Усилване означава да се използват по-дълги нотни стойности . Геометрически , това е разширяване между две позиции на оста.

$$\text{new position} = (\text{constant} > 1) * \text{position}, \quad \text{new note} = \text{note}$$

DIMINUTION - Намаляването използва по-кратки нотни стойности . Това е съкращаване или компресия между позициите на оста.

$$\text{new position} = (0 < \text{constant} < 1) * \text{position}, \quad \text{new note} = \text{note}$$

TO PLAY IN RHYTHM - Да се свири ритмично е когато се регулират нотите да попадат на точно време. Това също така се нарича квантазация.

$$\text{new position} = \text{INT}(\text{position}), \quad \text{new note} = \text{note}$$

TO PLAY IN DIFFERENT RHYTHM Смяна в ритъма или да се свири в различен ритъм е подобно на позиционно разширение или компресиране + квантазация.

$$\text{new position} = \text{INT}((\text{constant}) * \text{position}), \quad \text{new note} = \text{note}$$

REPETITION - Повторението означава да се копира последователността от събития.

$$\text{new position} = \text{position} + \text{constant}, \quad \text{new note} = \text{note}$$

Забелязва се че понятиятната система в математиката и музиката е подобна. Откриваме множество примери, свързани с тези прости математически оператори, в музиката, която свирим или слушаме.

4.2.7 ВЕРИГИ НА МАРКОВ

Друг метод за изследване на вероятностната система за създаване на алгоритмична музика са веригите на Марков. Веригите на Марков са вероятности системи, където вероятността за бъдещите събития зависи от едно или повече изминали събития. Марков анализира и изследва вериги от събития и проверява тенденцията на едно събитие да бъде последвано от друго. Използвайки тези анализи, редиците от случайни, но влезли във взаимоотношения събития, могат да бъдат генерирани. [30] Броят на миналите събития, които се взимат под внимание е известен като редът на веригата. Една верига, която взема само едно предшестващо събитие е от първи ред. N`

th ред във веригата на Марков, е представен от преходна матрица $n + 1$ размерност. Фигура 43 илюстрира такава преходна матрица за първи ред на веригата на Марков с четири възможни изхода: A, B, C, D. Миналите резултати се показват вертикално, а текущите се показват хоризонтално. Например за да се открие вероятността да се случи резултата Веднага след A в показаната матрица, откриваме A в първата колона (тя вече е минал резултат) и след това по хоризонтала се придвижваме до колона B. Появяването в този случай е 0.1 – има 1 на 10 шанс да се случи резултат B след резултат A.

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	0.2	0.1	0.3	0.4
<i>B</i>	0.5	0.1	0.2	0.2
<i>C</i>	0.5	0.2	0.1	0.2
<i>D</i>	0.2	0.2	0.3	0.2

Фигура 43. Пример за първа подредба на веригите на Марков [136]

Като пример, който да илюстрира употребата на веригата на Марков, за генериране на потоци от тонове, взимаме под внимание поредица от 8 тона, които съставляват До мажорна тоналност (C3 D3 E3 F3 G3 A3 B3 C4).



След това приемаме няколко правила, които са позволени на тоновете, за да следват даден тон:

ако е C3, тогава C3, D3, E3, G3, C4

ако е D3, тогава C3, E3, G3

ако е E3, тогава D3, F3

ако е F3, тогава C3, E3, G3

ако е G3, тогава C3, F3, G3, A3

ако е A3, тогава B3

ако е B3, тогава C4

ако е C4, тогава A3, B3

Всяко от тези правила представя преходните вероятности за следващия тон в поредицата. Например, след C3, всеки от петте тона C3, D3, E3, G3, C4 има 20% шанс да се случат, т.е. имат вероятност 0.2. В този случай вероятността е разпределена между 5 кандидата. Редът на вероятностните коефициенти в списъка, кореспондира на реда на елементите в редицата. Вероятностните списъци за всички тонове от До мажор могат да бъдат организирани в двумерна матрица, формирайки веригата на Марков (Фиг.44).

	C3	D3	E3	F3	G3	A3	B3	C4
C3	0.2	0.2	0.2	0.0	0.2	0.0	0.0	0.2
D3	0.33	0.0	0.33	0.0	0.33	0.0	0.0	0.0
E3	0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0
F3	0.33	0.0	0.33	0.0	0.33	0.0	0.0	0.0
G3	0.25	0.0	0.0	0.25	0.25	0.25	0.0	0.0
A3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
B3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
C4	0.0	0.0	0.0	0.0	0.0	0.5	0.5	0.0

Фигура 44. Пример за първа подредба на верига на Марков за тоновете от До мажор [136]

Веригите на Марков осигуряват много по-рафиниран механизъм за контрол на вероятностите. Използването на веригите на Марков, позволява на композитора да контролира редиците от събития, правейки вероятността на дадено събитие да зависи от предишното. [37] Връзката между събитията във веригите на Марков проявяват различни възможности, съобразно с това как техните вероятности са подредени в матрицата.⁵⁸ Ето и кратко описание на тези възможности:

- Достъпност ($e_x \rightarrow e_y$): едно събитие e_y е достъпно за друго събитие e_x , ако e_x може да бъде последвано от e_y .

- Способност за комуникация ($e_x \leftrightarrow e_y$): две събития си комуникират, ако и двете са взаимно достъпни. Комуникацията между събитията може да бъде **рефлексивна**, **възвратна** (ако едно събитие има достъп до самото себе си), може да бъде **симетрична** (реципрочна достъпност между които и да са две събития) или **преходна** (ако събитие e_x

⁵⁸ Jones, "Compositional Applications of Stochastic Processes", 47-48

си комуникира със събитие e_y и e_y си комуникира със събитие e_z , тогава e_x си комуникира с e_z).

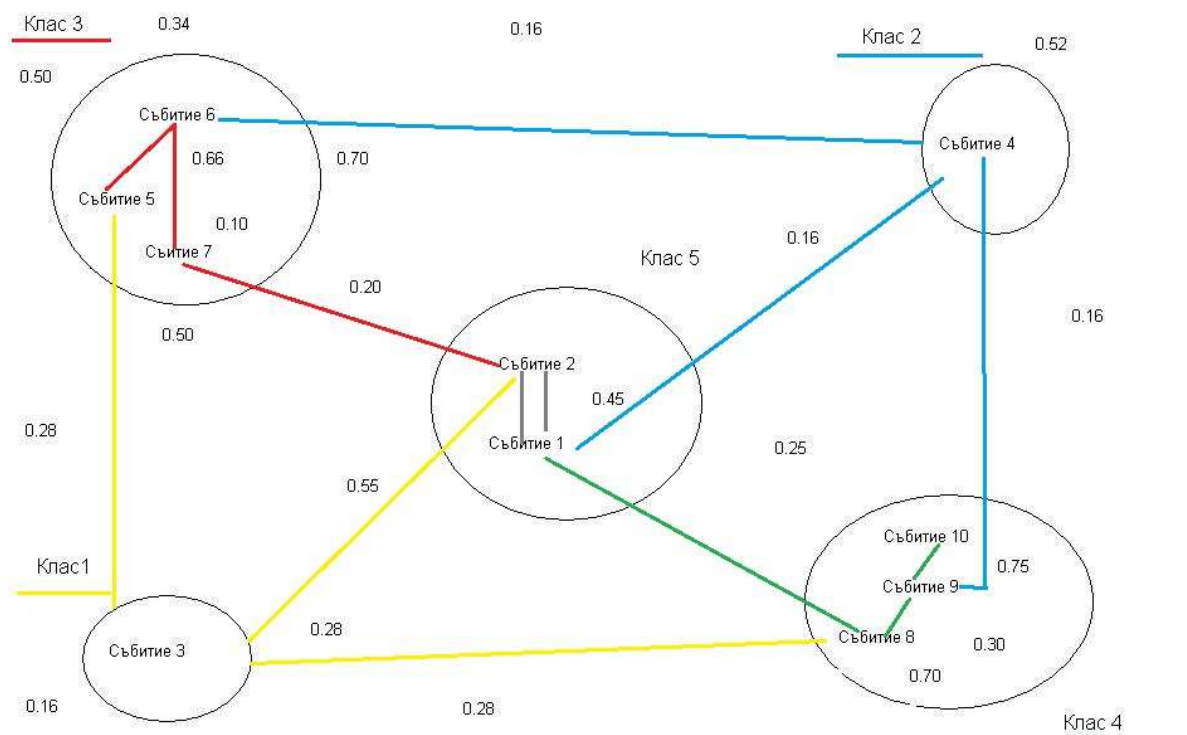
Събитията могат по-нататък да бъдат структурирани в еквивалентни класове от комуникиращи си събития. [37] Събития в един еквивалентен клас не могат да комуникират със събития в друг еквивалентен клас, но събития в един еквивалентен клас могат да бъдат достъпни от други събития в различен еквивалентен клас. Събития в еквивалентни класове могат да бъдат разделени в групи: **повтарящи се събития**, които са събития, определени да се появят в определен момент, след като вече са се появили веднъж и **преходни събития (неустойчиви, приходящи)**, са събития, които имат вероятност да не се появят. Ако едно събитие е повтаряемо, всички събития, с които то си комуникира в неговия клас са също така повтаряеми. Аналогично, всички събития, които си комуникират с преходно събитие, са също преходни. Следователно могат да бъдат дефинирани два типа еквивалентни класове: повтарящи се класове, които съдържат повтарящи се събития и преходни класове, които съдържат преходни събития. Събития от един повтарящ се клас нямат достъп до събития в други класове. Понеже събитията не могат да си комуникират между класовете, веднъж след като редицата от събития напусне един такъв преходен, приходящ клас, тя може повече да не се върне в него. Поради същата причина, ако една редица навлезе в един повтарящ се клас, може дълго да не я напусне. Повтарящите се класове формират ограничени редици. Веригите на Марков трябва да съдържат поне един повтарящ се клас, понеже не могат да се състоят само от преходни класове. Могат да бъдат дефинирани няколко типа вериги на Марков, съобразно типовете класове, които те съдържат:

1. Тип клас, който не може да бъде намален или съкратен: веригите на Марков съдържат само един повтарящ се клас и никакви преходни класове.

2. Ергодичен тип клас: веригите на Марков се състоят само от един повтарящ се клас и няколко преходни класа.

За да се изясни цялата тази теория и да се покаже как анализа на Марков се прилага в музикалното композиране, следва един практически пример:

Една верига на Марков с 10 музикални събития създава една подредена матрица. Диаграмата на връзката между събитията е показана на фигура 45.



Фигура 45. Диаграма на класове

Събитията са групирани в 5 класа: четири преходни(случайни), които са класове 1, 2, 3 и 4 и един повтарящ се клас (клас 5). Този пример е пример за ергодичен тип верига на Марков. Забележете също така как събитията 2, 3, 4, 6, и 7 показват рефлексивна комуникация. Кореспондиращата вероятностна матрица е както следва (Фиг.46):

Текущо събитие	Следващо събитие									
	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	.45	.55	0	0	0	0	0	0	0	0
3	0	.28	.16	0	.28	0	0	.28	0	0
4	.16	0	0	.52	0	.16	0	0	.16	0
5	0	0	0	0	0	.50	.50	0	0	0
6	0	0	0	0	.66	.34	0	0	0	0
7	0	.20	0	0	0	.70	.10	0	0	0
8	.25	0	0	0	0	0	0	0	.75	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	.70	.30	0

Фигура 46. Кореспондираща вероятностна матрица

Остава да бъде определено събитийното пространство. За този пример, се използват събития, за да се конструира композиция за соло цигулка (Фиг.47):

Фигура 47. Конструирание на композиция за соло цигулка [36]

Процедурата да се получи една редица от събития, съобразно вероятностната матрица е както следва:

1. Случайно се избира първото събитие от списъка със събития получавайки броя между 1 и броя на събитията.
2. Получава се случаен брой между 1 и максималния брой на събития, g .
3. В матрицата се проверява настройката за вероятност на текущото събитие на колоната g , P_{gr} .
4. Ако P_{gr} не е нула, тогава:
се получава случайно число между 0 и 1.

ако $n \leq P_r$ да се избере събитие в колоната r , като следващо събитие (Метода на Монте Карло).

ако $P_r = 0$ или $n > P_r$, тогава се връщаме към стъпка 2 (не се избира следващо събитие).

5. Повтаря се процедурата от стъпка 2, докато се постигне желания брой от събития в редицата.

Прилагайки тази процедура към нашето събитийно пространство, ще се генерират три възможни редици с 20 събития във всяка една:

$s1 = e7, e6, e5, e6, e6, e6, e6, e5, e6, e5, e7, e6, e5, e7, e6, e6, e5, e7, e2, e1$

$s2 = e3, e3, e8, e9, e10, e8, e9, e10, e9, e10, e8, e9, e10, e8, e1, e2, e2, e2, e2, e1$

$s3 = e4, e4, e4, e4, e4, e6, e6, e5, e7, e2, e2, e2, e2, e1, e2, e2, e1, e2, e2, e1$

Превеждайки ги в музика, се получават слените музикални последования (Фигури 48, 49 и 50):



Фигура 48. Музика от редица $s1$ [36]



Фигура 49. Музика от редица s2 [36]



Фигура 50. Музика от редица s3 [36]

Сравнявайки тези генерирани редици със събитийната диаграма на веригите на Марков показва ясно как различните класове се отразяват върху музикалния резултат.

За пример, веднъж като в един процес навлезе повтарящ се клас (в този случай клас 5, със съдържащи се събития 1 и 2), никога не го напуска. Музикално това създава повторение за събития 1 и 2.

Тези процеси са способни да генерират много интересни музикални откъси, но проблемът при тях е че в повечето случаи много бързо стават статични. А човешкият слух е устроен така, че възприема една музика като приятна, когато в нея има баланс между нивото на повтаряемост и новото звучене, които иска да изложи. Изборът на първоначални стойности за итерационния процес е много важен фактор, понеже процеси с много близки начални стойности могат да се изменят драстично след няколко итерации. Променяйки първоначалните стойности на една и съща итерационна музикална система, композиторът има възможност да създаде вариации в един пасаж, дори в цялата творба, която може да започне по много опростен начин, а в последствие да се разгърне. [37]

4.3 ТЕХНИКИ И ПРИЛОЖЕНИЕ НА МАТЕМАТИЧЕСКИ МОДЕЛИ В МУЗИКАТА

4.3.1 КОДОВО-ГРАФИЧНИ ПРЕОБРАЗУВАЩИ ТЕХНИКИ

Възпроизвеждането на числовия резултат от един алгоритъм към музикални събития се извършва чрез математическа операция, известна като кодово-графично преобразуване, изчертаване, мапване, кодиране. Кодово-графичното преобразуване се състои от създаване на 1:1 връзка между алгоритмичния числов резултат и поредица от установени музикални събития. Музикалните събития могат да бъдат отделни тонове, мотиви или дори цялостни музикални фрази. В повечето примери разглеждам музикални събития, които се асоциират само към тонове.

Основни се използват два типа техники за кодово-графично преобразуване: модулно-базирано (*modulo-based*) кодово-графично преобразуване и нормализирано (*normalized*) кодово-графично преобразуване.

Въпреки че повечето процеси могат да приложат тези два типа техники на кодово-графично преобразуване, някои се възпроизвеждат по-подходящо, използвайки определен тип техника за кодово-графично преобразуване.

4.3.1.1 МОДУЛНО-БАЗИРАНО КОДОВО-ГРАФИЧНО ПРЕОБРАЗУВАНЕ

Модулната операция е бинарна операция между цели числа. Тя връща остатък от делението на първото число на второто. Стандартният запис е $x \bmod y = n$. Например:

$$10 \bmod 3 = 1,$$

тъй като при деление на 3, числото 10 дава остатък 1.

Когато прилагаме този преобразуващ метод към един числов резултат на даден процес, x представя резултата на процеса, а y е общия сбор на елементите, в нашия установен ред от музикални събития. Резултатът n е винаги цяло число в интервала $[0, y - 1]$, което кореспондира на индекса на елемента в нашия ред от музикални събития. Модулно-базираното кодово-графично преобразуване най-добре се прилага в процеси, в които числовия резултат нараства без ограничение, подобно на числовите поредици. За процеси, в които числовия резултат нараства в много малки интервали (подобно на изчислителните уравнения), модулното кодово-графично преобразуване е напълно непрактично, понеже извежда в сумата само интегралната част от резултата на процеса. В тези случаи най-добрият тип на кодово-графично преобразуване е нормализираното кодово-графично преобразуване.

4.3.1.2 НОРМАЛИЗИРАНО КОДОВО-ГРАФИЧНО ПРЕОБРАЗУВАНЕ

Този преобразуващ метод използва следната формула, за да направи връзка между алгоритмичните данни и установените елементи в редицата от музикални събития :

$$\text{Event No.} = \left\lfloor \frac{(\text{value} - \text{minval}) * \text{numevents}}{\text{maxval} - \text{minval}} \right\rfloor$$

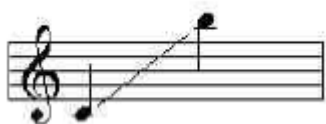
Където *value* е текущата стойност на процеса, която е преобразувана; *minval* е минималната стойност в реда на данните от процеса; *maxval* е максималната стойност, а *numevents* е броя на елементите в нашия ред от събития. Скобите $\lfloor \rfloor$ посочват да се извежда само интегралната (целите числа) част от резултата, игнорирайки всички десетични дробни. Методът за нормализирано кодово-графично преобразуване

ефективно преобразува реда от данни в интервала [0, 1]. Минималните и максималните стойности в реда на данните от процеса трябва да бъдат изчислени предважно. Този преобразуващ метод е по-подходящ за процеси, чийто резултат е ограничен в малки интервали. Като допълнение, този метод преобразува много по-правдоподобно контура (начина, по който се държат числовите данни) в реда на музикалните събития.

За да се посочат разликите между тези две преобразуващи схеми, е посочен пример. Първо се преобразува едно изчислително уравнение, използвайки и двете техники на кодово-графично преобразуване. Математическата формулировка на това уравнение е както следва :

$$x_{n+1} = \mu x_n (1 - x_n)$$

Резултатът от изчислителното уравнение е ограничен в интервала [0, 1]. Музикалното пространство от събития ще обхваща тоновете от хроматичната гама от C₄ до B₅ (24 събития), (Фиг.51) .



Фигура 51. Пространство на тоновете събития

След това се генерира една редица от 10 събития, използвайки стойност $\mu = 3.57$.

Числовият резултат на уравнението е както следва:

0.892500	0.342519	0.803963	0.562655	0.878486
0.381093	0.842024	0.474880	0.890247	0.348814

Прилагайки модулното кодово-графично преобразуване към стойностите горе, се постига следния резултат :

0 0 0 0 0 0 0 0 0 0

което възпроизвежда следната тонова последователност (Фиг.52):



Фигура 52. Модулно базирано кодово-графично преобразуване [98]

Използвайки метода на нормализирано кодово-графично преобразуване се получава следната редица, като максималните и минималните стойности на алгоритъма са 1 и 0, респективно:

21 8 19 13 21 9 20 11 21 8

което възпроизвежда следната тонова последователност (Фиг.53):



Фигура 53. Нормализирано кодово-графично преобразуване [98]

Както ясно може да се види, метода на нормализирано кодово-графично преобразуване е по-подходящ за този процес, в конкретния случай. Понеже всеки процес има различно поведение, трябва първо да се анализира числовия резултат, за да се реши кой преобразуващ метод е по-подходящ. Някои процеси могат да бъдат преобразувани само като се използва един от двата метода, докато някои други дават много задоволителни резултати, без значение дали използват единия или другия метод. В крайна сметка, решението е изцяло в ръцете на композитора.

4.3.2 СТОХАСТИЧНИ АЛГОРИТМИ (*STOCHASTIC ALGORITHMS*)

Математически, стохастичните процеси се дефинират като редица от случайно разпространени количества. Стохастичните формули се използват от статистите на данни, за да се определят образци, които са по-съвместими с данните, които могат да се достигнат.⁵⁹ В музиката, стохастичните процеси се използват обратно, т.е. те се грижат да осигурят структурни рамки за синтез на колекция от музикални събития. [37]

Стохастичните музикални събития изискват случайни редици от числа. Случайността понякога е много трудна едновременно да бъде определена

⁵⁹ K.Jones, "Compositional Applications of Stochastic Processes", Computer Music Journal 5(2) 1981, 45

математически и след това да бъде пресъздадена от компютъра. В компютъра нищо не може да бъде наистина случайно – оставено на произвола; компютърът симулира случайността чрез сложни математически процедури и следователно генерираните редици от случайни числа са наистина “псевдо-случайни”. Процесът на генериране на случайни числа в компютъра всъщност е напълно детерминиран. Редиците от случайни числа изглеждат като малки порции от обширни кръгове от числа, които се повтарят след определен брой итерации. Всяко ново число в редицата се създава чрез прилагане на математическа трансформация върху предишното число. Случайността се грижи за това да определи откъде точно да започне в редицата. Идентични стойности генерират идентични редици от псевдо-случайни числа.

В повечето примери, които използват генерирането на случайни числа, се използва най-добрата функция-генератор на случайни числа. Тази функция генерира редица от случайни числа с период на повторение по-голям от 10^{18} . Когато се извиква, функцията която ще се нарича просто с буквата R, връща случайно число в интервала 0,1. [73]

Кодово-графичното преобразуване на редица от случайни числа към музикални събития е ясна процедура. Започва с разглеждането на събитието, на което трябва да бъде приложено кодовото преобразуване. Елементите от тази колекция могат да бъдат която и да е музикална структура, от индивидуални тонове, кратки мотиви до дълги завършени фрази. За по-голямо удобство, следващия пример ще засегне само тонове. Композиторът първо трябва да реши тоновия обхват, който да служи като поредица тонове, която ще бъде преобразувана от функцията за случайни числа. Това може да бъде която и да е колекция от тонове, например, хроматична гама от две октави в диапазона от C_4 до B_5 .

Тази поредица съдържа 24 тона (височини), номерирани съответно от 0 за C_4 до 23 за B_5 .

Функцията R, която генерира случайни числа, извежда псевдо-случайни числа между (0,1). Докато резултата от функцията се ограничава в малък интервал, най-подходящия преобразуващ метод е нормализиращото кодово-графично преобразуване. Понеже минималните и максимални стойности на функцията са респективно 0 и 1, преобразуващата формула е еквивалентна на :

$$\text{Event No.} = [R * 24]$$

Резултатът ще бъде винаги число между 0 и 23, което е част от диапазона на нашата тонова поредица. Нулата съответства на C_4 , 1 на $C\sharp_4$, 2 на D_4 и т.н.

Прилагайки тази процедура 128 пъти върху избраната поредица от тонове, се получава следната поредица от тонове (произволно определени с продължителност шестнадесетина), (Фиг.54):



Фигура 54. Стохастични тонове [137]

Всяка поредица от музикални параметри ще бъде преобразувана подобно на примера по-горе. В този конкретен пример, всички тонове в поредицата са еднакво вероятностни, т.е. те имат една и съща вероятност да се случат. Това е ефект, еквивалентен на този при който се използва зар с 24 страни, чрез който се избират тоновете. Ако всички елементи в тоновата поредица имат същата вероятност, то наблюдаваме един алеаторен процес, в който всички елементи имат една и съща вероятност да се появят. В алеаторните процеси, достатъчно дълги редици от генерирани елементи ще покажат незабележими образци (Фиг.54). Една допълнителна стъпка се състои от извеждане вероятността за всеки елемент в тоновата поредица. За всеки елемент се определя като “вероятностен да бъде избран”. По този начин наблюдаваме една преценена редица, където елементите с по-висока вероятност ще имат и по-голяма възможност да се случат. За пример, два параметъра, височина и продължителност, ще бъдат преобразувани. Тоновата поредица ще бъде хроматична гама от една октава в диапазона от C_4 до B_4 . Нека поредицата от тонови трайности, бъде както следва : цяла нота, половин нота, четвъртина, осмина, шестнадесетина.

Следващата стъпка е да се определят следните вероятности към последователните членове на тоновите поредици (Таблицы 3 и 4):

Елемент	Вероятност	Индекс
C	.17	0
C#	.02	1
D	.07	2
D#	.03	3
E	.15	4
F	.12	5
F#	.01	6
G	.16	7
G#	.08	8
A	.10	9
A#	.02	10
B	.07	11

Таблица 3. Вероятностни членове на тоновата поредица

Елемент	Вероятност	Индекс
Whole	.03	0
Half	.06	1
Quarter	.11	2
Eighth	.34	3
Sixteenth	.46	4

Таблица 4. Вероятностни ритмични стойности

Вероятностите могат да придобият която и да е стойност между 0 и 1. Сумата от вероятностите на всички членове винаги ще е равна на 1. Един елемент *e* с вероятност от да кажем, .20, означава че 20% от генерираната редица от събития са подобни на елемента *e*. С други думи, в достатъчно дълги редици, елемент *e* ще се появи в 20% от

времето. Добавяйки и вероятността, се въвежда една нова стъпка в преобразувания процес, който е както следва :

1. Изтегляне се случайно число r между $[0,1)$

2. Умножава се r по броя елементи в нашата поредица от тонове и се извежда интегралната (цялата) част от резултата: $[r*12]$. Това предоставя стойността n между 0 и 11, която съответства на един от елементите от редицата.

3. Откриване на вероятностната стойност (P_n) за елемента n в редицата.

3.1. Изтегляне на ново случайно число r_2

3.2. Дали $r_2 \leq P_n$? Ако отговора е положителен, тогава целия процес продължава, в другия случай се връща към стъпка 1.

4. Елементът се приема.

5. Повтаряне на стъпки от 1 до 4 за поредицата от тонови трайности (в нашия случай 5 елемента) и присъединяване на трайността към предишния селектиран тон.

Всичко се повтаря, докато се изберат желаните брой елементи.

Стъпка 3.2. е ключовата в този пример. Понеже всички възможни стойности, генерирани от генератора на случайни числа са еднакво вероятностни, вероятността за всяко случайно число да бъде равно или по-малко от вероятността на един елемент, който е бил избран, е еквивалентна на вероятността на елемента, който е бил избран. Представено символично:

$$r_2 \leq P_n \rightarrow n$$

Този процес, известен като метода Монте Карло, може да се използва да се определи дали един елемент е селектиран или не.

Прилагайки този метод към избраната тонова поредица, се извежда следната музикална поредица (Фиг.55):



Фигура 55. Музикална поредица от метода Монте Карло [137]

Ясно е че елементите с висока вероятност (като тонове C, E и G и стойностите шестнадесетини) се появяват по-често от елементите с по-ниска вероятност. За пример, в тази музикална поредица от 50 елемента, тонът C се появява 9 пъти или 18%, в съответствие с неговата вероятност. По същия начин, има 24 шестнадесетини (48% от нотните стойности), което също е в съответствие с неговата вероятностна стойност.

4.3.3 МУЗИКА ОТ ХАОСА

Хаотичните системи могат недвусмислено да се категоризират в два класа: консервативен, в който енергията е запазена и разпиляващ клас, в който енергията се излъчва към обкръжаващата среда.⁶⁰ Мнозинството от естествени феномени, представени от тези системи са от разпиляващ клас. Поради това че разпиляващите системи губят енергията си непрекъснато, тяхното фазово пространство е пространствената локация, където системата се разгръща и се трансформира към определено състояние (обикновено преминавайки през първоначално преходно положение), известно като атрактора на системата. [22] Атракторът на системата може да бъде една единствена точка (ограничен-точков атрактор), подобно на движението на махалото; може да бъде редица от точки (кръгов, цикличен атрактор), между които

⁶⁰ R.Bidlack, "Chaotic Systems as Simple (but Complex) Compositional Algorithms, "Computer Music Journal 16(3) (1992), 33

орбитите на системата трептят, вибрират; могат да бъдат сложна форма, обикновено с фрактална измерност (хаотичен или странен атрактор). Фазовото пространство на консервативните системи – винаги е константа.

Математическите системи, които моделират хаос са едновременно итерационни системи, формулирани с нелинейни различаващи се уравнения, в които орбитите от атрактора се състоят от дискретни точки или могат да бъдат непрекъснати потоци, формулирани с нелинейни диференциални уравнения, в които орбитите са непрекъснати, цели криви. [22]

4.3.3.1 ЕДНОМЕРНИ ХАОТИЧНИ СИСТЕМИ

Възпроизвеждайки хаотичните системи към музика предполага конструирането на кодово-преобразуващ процес между математическото описание на системата и музикалните параметри. Докато хаотичните системи са моделирани с нелинейни уравнения, процесът се състои от кодово-графично преобразуване на цифровия резултат на тези уравнения към музикално събитийно пространство.

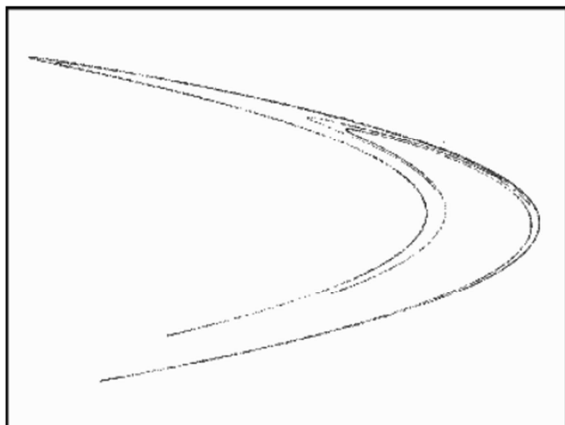
4.3.3.2 АТРАКТОРИ ОТ ДВУМЕРНИ ХАОТИЧНИ КАРТИ

В този раздел ще разгледам три двуизмерни итерационни карти и как те се преобразуват към музикални събития. Тези атрактори са Henon Map, атрактора Gingerbread Man и атрактора Barry Martin.

Атрактор Henon Map се именува след като бива открит от Майкъл Хенон, астроном от Обсерваторията в Ница, Франция. Представява хаотична орбита в две измерения. Въпреки че е направен изцяло от линии, орбитите на този атрактор не се движат непрекъснато, а преминават от една точка на друга в атрактора. **Henon атрактора** има безкрайна структура. Дефинира се математически чрез следното итеративно уравнение:

$$\begin{aligned}x_{n+1} &= 1 + y_n - ax_n^2 \\y_{n+1} &= bx_n \\x_0 = y_0 &= 0, a = 1.4, b = 0.3\end{aligned}$$

Когато се изчертае това уравнение изглежда по следния начин (фиг.56):



Фигура 56. Henon атрактор [138]

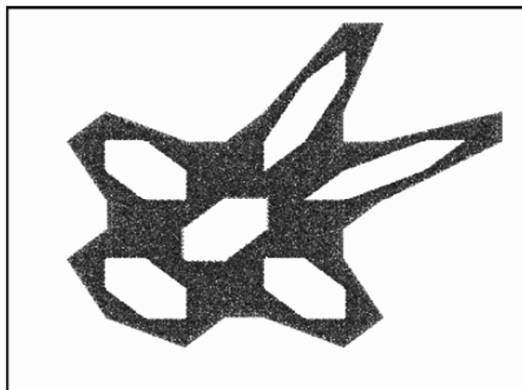
Gingerbread Man атрактор е друга двуизмерна итеративна карта. [60]
Предложен е от Р.Девани през 1988 г.⁶¹ Определя се от следната рекурентна връзка:

$$x_{n+1} = 1 - y_n + |x_n|$$

$$y_{n+1} = x_n,$$

където $x_0 = -0.1, y_0 = 0$.

Когато се изрисува този атрактор изглежда по следния начин (Фиг.57):



Фигура 57. Gingerbread Man атрактор [139]

Най-накрая, **Martin атрактор**, предложен от Бари Мартин от Университета Астън в Бирмингем, Англия, а за първи път се разисква от А.К. Дюдни през 1986 г., е двуизмерна орбита, дефинирана от следните рекурентни връзки:

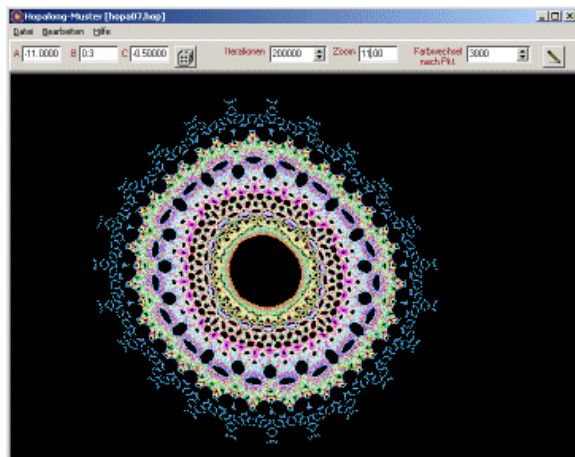
$$x_{n+1} = y_n - \sin x_n,$$

$$y_{n+1} = a - x_n,$$

⁶¹ H.O.Peitgen and D.Saupe, eds., The Science of Fractal Images (Berlin:Springer, 1988), 149.

$$y_0 = x_0 = 0,$$

където a е контролиращия параметър. При $a = \pi \approx 3.1415926$, атрактора **Martin** изглежда по следния начин (Фиг.58):



Фигура 58. Атрактор Martin [140]

Всички процеси на кодово-графично преобразуване показани дотук, използват едно и също събитийно пространство: еднооктавова хроматична скала от C_5 до B_5 .

Понеже това са двуизмерни итеративни карти, от всяка итерация на уравнения, ние получаваме нова стойност за x_n и y_n . Това позволява различни преобразуващи алтернативи.

Една от тях е да се комбинират двете координати в една стойност: разстоянието от началото на координатите до точката (x_n, y_n) в атрактора, а именно $\sqrt{x_n^2 + y_n^2}$.

Във финалното превеждане към музика, повтарящите се тонове са легатирани. Модулно-базираното кодово-графично преобразуване се използва във всички следващи примери.

Едно музикално последование от 160 събития (тонове) се генерира от всеки атрактор (Фиг.59, 60, 61), като шестнадесетините са избрани произволно за трайност на тоновете в този случай:



Фигура 59. Музыка от Атрактор Непон [141]



Фигура 60. Музыка от Gingerbread Man аттрактор [142]



Фигура 61. Музика от Атрактор Martin [143]

Следващото проучване на тези мелодии ще покаже, че те не са толкова случайни, колкото изглеждат на пръв поглед. Първо се изчислява честотата на всеки тон:

Музика от атрактор Henon:

Височина	Брой появявания	Отклонение
C	15	+2
C [♯]	8	-5
D	12	-1
D [♯]	18	+5
E	20	+7
F	13	0
F [♯]	8	-5
G	23	+10
G [♯]	13	0
A	7	-6
A [♯]	5	-8
B	18	+5

Музика от Gingerbread Man атрактор:

Височина	Брой появявания	Отклонение
C	6	-7
C [♯]	11	-2
D	9	-4
D [♯]	13	0
E	16	+3
F	19	+6
F [♯]	20	+7
G	17	+4
G [♯]	16	+3
A	11	-2
A [♯]	11	-2
B	11	-2

Музика от Атрактор Martin:

Височина	Брой появявания	Отклонение
C	26	+13
C [♯]	20	+7
D	13	0
D [♯]	22	+9
E	12	-1
F	15	+3
F [♯]	4	-9
G	11	-2
G [♯]	9	-4
A	4	-9
A [♯]	12	-1
B	12	-1

Очевидно е от горепосочените примери, че някои тонове преобладават над други тонове. В една наистина случайна колекция от тонове, всички тонове ще имат същата честота. Дали тази появяваща се честота ни казва нещо относно тяхното вероятностно разпределение? Нека да ги препоредим по друг начин:

Музикално последование от атрактор Henon:

Честота	5	8	12	13	15	18	23	20	18	13	8	7
Височина	A [♯]	C [♯]	D	G [♯]	C	B	G	E	D [♯]	F	F [♯]	A

Музикално последование от атрактор **Gingerbread Man**:

Честота	6	11	11	13	16	19	20	17	16	11	11	9
Височина	C	C#	A	D#	E	F	F#	G	G#	A#	B	D

Музикално последование от атрактор **Martin**:

Честота	4	11	12	15	20	26	22	13	12	12	9	4
Височина	F#	G	B	F	C#	C	D#	D	A#	E	G#	A

Забележимо е че всичките три последования проявяват тип на вероятно разпределение.

Също така, колекцията от преобладаващи тонове (тези, които са с по-голяма вероятност да се случат) е различна във всяко едно от последованията (и ще бъде различна в което и да е музикално последование, получено от различен атрактор). Това придава уникална характеристика на всяко музикално последование. Атракторите от които са получени тези последования показват високо степенна структура, която впоследствие може да бъде трансформирана към тяхната музикална транскрипция.

Вглеждайки се в музикалното последование, генерирано от **Henon** атрактор, откриваме че определени образци от тонове се повтарят (Фиг.62):





Фигура 62. Музикално последование генерирано от Непон атрактор [141]

Забелязва се, че образците не повтарят абсолютно едно и също. Няколко тонове може да се променят, но техния контур е сходен.

Музикалното последование, получено от атрактор Martin също показва образци на повтарящи се тонове (Фиг.63):



Фигура 63. Музикално последование, получено от атрактора Martin [143]

Същото се наблюдава и в музикалното последование на атрактора **Gingerbread Man** (Фиг.64):



Фигура 64. Тактове 9, 2, 3, 4, края на 3 и началото на 4 такт [142]

Забележете тези малки клетки, които се появяват в инверсия, движещи се в обратна посока. Има дори пример на последование от тонове, което не върви назад, а върви в прогресия напред (Фиг.65):



Фигура 65. Края на 5 такт, началото на 6 [142]

Интерваловите връзки между тоновете също са различни в трите примера. В музикалното последование получено от **Henon** атрактор изглежда, че има сродство между малки и по-големи интервали (секунди, терци и квари). Това всъщност прилича на някои от визуалните интерпретации на атрактора, който се състои от линии, разделени на различни разстояния.

В **Martin** **атрактор** изглежда, че преобладават по-големите интервали (кварта и най-вече сексти), правейки по този начин по-отчетлива мелодична линия.

В музикалното последование на атрактор **Gingerbread Man** преобладават по-малките интервали (малки и големи секунди, малки терци), а мелодичната линия е свързана.

В едно случайно последование всички тонове, интервалови съотношения, ще имат една и съща вероятност да се случат и няма да има изявен образец.

Друг начин да се преобразуват двуизмерни карти е като се преобразува всяка координата към различен музикален параметър (като височина или ритъм). Може да се преобразува, например, X координатата към височинно събитийно простространство, а Y координатата към ритмична редица или да се използва същата координата, за да се преобразуват и двете събитийни редици.

За да се демонстрират тези алтернативни начини на кодово-графично преобразуване, ще се преобразуват същите три итеративни карти в две отделни събитийни пространства (височина и ритъм), използвайки стойностите на X и Y координатите. Тоновият диапазон ще бъде хроматична скала от две октави, от C₄ до B₅ (24 елемента). Ритмичното събитийно пространство ще съдържа 5 стойности (16 (шестнадесетина), 8 (осмина), точкувана осмина, 4 (четвъртина) и 4 легатирана с 16)). В този случай ще се използва нормализиращо кодово-графично преобразуване. За да се

приложи този тип на кодово-графично преобразуване, максималните и минималните стойности на X и Y се изчисляват предварително, в редица от 10 милиона итерации от уравненията. Ето как изглеждат генерираните музикални уравнения (Фиг.66, 67, 68):



Фигура 66. Музика от атрактора **Henon**. Височината е преобразувана към X координатата, а продължителността към Y координатата [141]



Фигура 67. Музика от атрактора **Henon**. Височината е преобразувана към Y координатата, а продължителността към X координатата [141]



Фигура 68. Музика от Martin атрактор. Височината е преобразувана към X координатата, а продължителността към Y координатата [143]

Всеки атрактор генерира различни последования, въпреки че всички се преобразуват към едно и също събитийно пространство. Интересно е да се отбележи, че в същия атрактор, две различни преобразуващи схеми (височина преобразувана към X и продължителност към Y и обратното) генерират много сходни последования.

4.3.3.3 АТРАКТОРИ ОТ ТРИИЗМЕРНИ ХАОТИЧНИ КАРТИ

Триизмерните атрактори съществуват в триизмерното пространство. Техните уравнения съответно имат три променливи. Всяка точка в атрактора е изразена от три координати (X, Y, Z). Всяка от тези координати може да бъде преобразувана към различно музикално пространство от събития (като височина, ритъм и динамика) по подобен начин, както това става в двуизмерните атрактори. Като пример, ще наблюдаваме два триизмерни атрактора: атрактор **Rössler** и **Pickover** атрактор.

Pickover атрактора е една итеративна (повтаряща се) карта в триизмерността, предложен от Клифърд Пикоувър.⁶² Формулира се от следните уравнения:

$$x_{n+1} = \sin(ay_n) - z_n \cos(bx_n),$$

$$y_{n+1} = z_n \sin(cx_n) - \cos(dy_n),$$

$$z_{n+1} = \sin x_n,$$

където a, b, c и d са управляващите параметри. Стойностите на тези параметри по принцип са $a = 2.24, b = 0.43, c = -0.65$ и $d = -2.43$.

Когато се изчертае тази функция генерира следния атрактор (Фиг.69):



Фигура 69. Pickover атрактор [144]

За съжаление е загубено едно измерение, понеже образът е отпечатан в двуизмерна хартиена повърхност. Всяка координата X, Y, Z се преобразува към три различни музикални поредици от събития: поредица от височини (хроматична гама в две октави в диапазона от C_4 до B_5); една ритмична поредица (шестнадесетини, осмини, точкувани осмини, четвъртини и четвъртини точкувани с шестнадесетини); поредица от динамика (ff, mf, mp, pp). [54] В преобразуващата схема височината се преобразува към Y координата, ритъмът към X координата, а динамиката към Z координата. Както и при двуизмерните атрактори, максималните и минимални стойности за X, Y и Z са се изчисляват от редица от десет милиона итерации на уравненията. Изходната музикална поредица е както следва (Фиг.70):

⁶² C.A.Pickover, Million – Point Sculptures, Computer Graphics Forum 10(4) (1991) : 333-336



Фигура 70. Музика от Pickover атрактор. Височината е преобразувана към Y координата, ритъмът към X координата, а динамиката към Z координата [36]

Динамиката е представена от 4 линии с различна дължина, показани под самите ноти. Най-дългата линия съответства на *ff*, а най-късата на *pp*.

Прехвърляйки различните координати към събитийните редици, могат да бъдат приложени различни преобразуващи схеми към един и същ атрактор. В следващата преобразуваща схема на същия атрактор, височината е прехвърлена към X координата, ритъмът към Z, а динамиката към Y (Фиг.71).



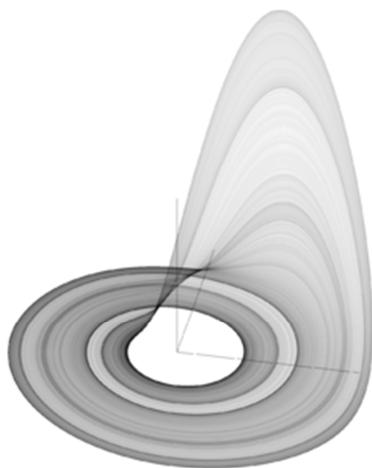
Фигура 71. Музика от Pickover атрактор. Височината е преобразувана към X координата, ритъмът към Z, а динамиката към Y [36]

Нужно е да се отбележи, че като опозиция на преобразуващите схеми на атракторите Henon, Martin и Gingerbread, които разгледах по-рано, в Pickover атрактора различните координати не дават същите стойности на случващи се събития. При Pickover атрактора също така липсва симетрията, която имат останалите три атрактора. [61] Това не означава, че музикалното кодово-графично преобразуване от Pickover атрактора е лишено от структура. Става ясно от фигурите на примери 70 и 71, че Pickover атрактора произвежда строго организирани последования.

АТРАКТОР RÖSSLER, предложен от Ото Рьослер, е непрекъснато движение (поток) в триизмерността.⁶³ Тясно свързан е с атрактора Lorenz (Лоренц), който е математически модел на система за времето, развита от Едуард Лоренц в MIT. Атрактора RÖSSLER е дефиниран от следните диференциални уравнения:

$$\begin{aligned}x_{n+1} &= x_n - y_n dt - z_n dt, \\y_{n+1} &= y_n + x_n dt + a y_n dt, \\z_{n+1} &= z_n + b dt + x_n z_n dt - c z_n dt, \\x_0 &= y_0 = z_0 = 1, \\dt &= 0.04, a = 0.2, b = 0.2, c = 5.7.\end{aligned}$$

Въпреки че непрекъснати движещи се системи като тази се характеризират с непрекъснати орбити, този факт трябва да бъде доближен в компютъра от орбити на дискретни точки, разделени във времето от малка сума, dt . [48] Това е постигнато с решаването на различни уравнения по метод известен като числова интеграция.⁶⁴ Когато се изчертае, атрактора RÖSSLER изглежда по следния начин (Фиг.72):



Фигура 72. Атрактор RÖSSLER [145]

По този начин са преобразувани всички параметри към един глас. Може също така да се преобразува всяка координата към различен глас, като това ще създаде полифонична структура (тази техника може да бъде приложена към двуизмерни атрактори). Музикалният превод на атрактора RÖSSLER преобразува всяка координата (X, Y, Z) към три различни събитийни редици (подобно на Pickover атрактор) в три различни гласа.[22]

⁶³ Lorenz, Deterministic non-periodic flow, 130-141

⁶⁴ Bidlack, Chaotic Systems as Simple (but Complex) Compositional Algorithms, 40

В глас едно, височината се преобразува към X координата, ритъмът към Y и динамиката към Z ; в глас две, височината се преобразува към Y , ритъмът към Z , а динамиката към X ; третият глас преобразува височината към Z , ритъмът към X , а динамиката към Y . Изходният резултат е показан на фигури 73 и 74:



Фигура 73. Музика от атрактора RÖSSLER [36]



Фигура 74. Музика от атрактора RÖSSLER (продължение) [36]

Забележете как непрекъснатата природа на орбитите на атрактора се реализират музикално като възходящи и низходящи гами, с различно темпо и динамика. Полифоничното кодово-графично преобразуване произвежда слухово усещане, което рефлектира по-близо до пространствената природа на тези атрактори. По-големи мерни системи предлагат повече степени на свобода в преобразуващите процеси. Една четиримерна хаотична система, като системата Henon – Heiles, дава четири променливи, които могат да бъдат преобразувани към четири различни събитийни редици, като височина, трайност, динамика и тембър. Начинът, по който тези по-високи мерни системи ще бъдат преобразувани не се различава от по-ниските класове мерни системи.

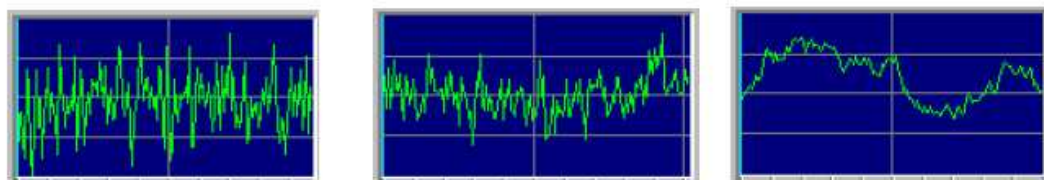
4.3.4 АЛГОРИТМИ ЗА ПОЛУЧАВАНЕ НА ШУМ

Шумът също може да бъде източник за алгоритмично композиране. Шумът може да бъде категоризиран в три основни типа: бял, розов и кафяв. Те се разграничават по това как техния енергиен спектър (измерван в dB) варира като функция на честотата (измервана в Hz). Енергията (силата) на белия шум се разпределя равно върху всички честоти. Неговият енергиен спектър $P(f)$ като функция на честотата f прилича на: $P(f) = 1/f^a$, където степента a е 0. За пример, белият шум на семплираща честота от 44,100 Hz има повече сила между 100 и 600 Hz, отколкото между 20,100 и 20,600 Hz. Честотите в белия шум са изцяло несъпоставими и несвързани. Белият шум е това, което чуваме като постоянен шум по радиото. [76]

Кафевият шум се намира точно на обратната страна – неговият честотен спектър е силно свързан. Неговият енергиен спектър $P(f)$ е близък до $1/f^2$.

$1/f$ шума или по-известен като розов шум, чийто енергиен спектър е $1/f$ (степента a в този случай е много близка до 1), е нещо средно между белия и кафевия шум. $1/f$ шума има равно разпределение на енергията, когато честотата се преобразува по логаритмична скала. Сумата от енергия е една съща както между 100 и 500 Hz, така и между 1000 и 5000 Hz. Честотите в розовия шум не са свързани, както са при кафевия шум, нито са случайни както са при белия. За нашия слух, розовия шум звучи естествено, дори приятно. Всъщност, розовия шум съществува навсякъде в природата: звукът на водопада; звукът на дъжда; звукът, който издават пчелите в пчелния кошер – всичко това е $1/f$ шум. [75]

Една графика на честотната плътност на тези шумове (Фиг.75) помага да се визуализират разликите между тях:



Фигура 75. Спектър на бял, кафяв и розов шум [75]

През 1975 г. Ричард Вос (*Richard Voss*) и Джон Кларк (*John Clarke*) в Университета на Калифорния анализират няколко записа на музика и говор.⁶⁵ В този

⁶⁵ J. Clarke and R. F. Voss, “ $1/f$ noise in music and speech”, *Nature* 258 (1975), 317-318

анализ те заключават, че аудиото силата на музиката, например силата предавана от говорителите, е много близка до тази на $1/f$ шума. Тяхното изследване разглежда много различни типове музика, от Бранденбургски концерт за пиано рагтайми на Скот Джоуплин (*Scott Joplin*). Резултатите от това изследване става повод за разгорещени спорове между изследователите, понеже тези записи са били обединение на творби от различни стилове музика и различни композитори, редактирани заедно в последование, придружено от говор и коментари. Аргумента срещу това изследване е бил че това обединение на различна музика и звуци не представя инстинно музиката в нейната есенционна форма, като една непрекъсната творба.

Жан-Пиер Бун (*Jean-Pierre Boon*) и други изследователи формулират различна техника за количествен анализ на музиката.⁶⁶ Вместо да се анализират записи на музика, един синтезатор се свързва с компютър, и композицията бива просвирена от синтезатора от даден изпълнител. Творбите дигитално се съхраняват в компютъра, което включва дискретизация на височината и продължителността, след което се използва обработка на данни, за да се конструира фазов портрет на музиката. Използвани са 19 класически творби от всички музикални епохи и четири джаз творби. Други два типа последования се подлагат на тест за сравнение с творбите: повтарящи се възходящи и низходящи гами и последование от 5000 ноти базирани върху алгоритъм на бял шум. [17]

Анализът на тези творби по този начин предоставя интересни резултати. Когато са изчислили стойностите a за спектралната плътност на творбите $1/f^a$ резултатите са се различавали значително от резултатите постигнати от Вос и Кларк. По този начин a изглежда е варирано между 1.79 и 1.97, което е по-близко до кафявия шум, отколкото до розовия [75]. Разликата се обяснява с употребата на отделни откъси музика, отколкото на дълги и разтегнати фрази.⁶⁷

Алгоритмите за получаване на музика от бял и кафяв шум са забележително прости. “Бялата” музика може лесно да бъде създадена от просто кодово-графично преобразуване на резултата от генератор на случайни числа към колекция от събития в подредено събитийно пространство. Пример, който симулира употребата на кафяв шум

⁶⁶ J.P. Boon and O. Decroly, “Dynamical Systmes Theory for Music Dynamics,” *Haos* 5(3) (1995), 501-508

⁶⁷ Както и самия Бун казва, “Ако анализа на музикалната динамика се възприема като процедура, която идентифицира и характеризира елементите на музикалната същност, то отделната творба е общоприет разпознаваем обект, който трябва да бъде изследван. В този смисъл, значението на дългите и разтегнати последования от различни музикални творби, е неясен.”

е веригата на Марков. Споменаваме отново този тип верига на Марков, при който се създават свързани последования от събития, които симулират кафявия шум. 1/f шума се оказва, че е много по-сложен за генериране и кодово-графично преобразуване на музикални събития.

Вос и Кларк са първите изследователи, които експериментират, за да генерират музика от 1/f шума. Това е и първия метод, състоящ се в генерирането на 1/f шум чрез електронни средства. Електрическото напрежение генерирано чрез тази процедура е семплирано, квантазирано и преобразувано към серии от числа, чийто спектрална плътност е била с тази на 1/f шума. След това тези числа са били преобразувани към поредици от височини (хроматична скала от две октави) и към продължителност. Те също така изобретяват алгоритмичен метод, чрез който да симулират спектралната плътност на 1/f шума.

През 1978 г. Мартин Гарднър (*Martin Gardner*) описва прост алгоритъм включващ три зара, които емулират спектралната плътност на 1/f шума.⁶⁸

Едно нелинейно уравнение, чийто изходен резултат наподобява 1/f шума, бива предложено от М. Шрьодер (*M.Schroeder*):

$$x_{n+1} = \mu x_n + r\sqrt{1 - \mu^2},$$

където μ е кое да е число в интервала (0, 1), а r е случайна стойност избрана за всяка итерация на уравнението. Стойността на μ определя качеството на изходния резултат в отношение към реалния 1/f шум. Това което следва от уравнението, предложено от Шрьодер, са три музикални примера, получени от бял, кафяв и 1/f шум. [26] Събитийното пространство е едно и също за всички (Фиг.76, 77, 78): хроматична скала в две октави от C₄ до B₅.



Фигура 76. Музика от бял шум [75]

⁶⁸ M. Gardner, "White and Brown Music, Fractal Curves and 1/f Fluctuations," Scientific American (April 1978): 16-32



Фигура 77. Музика от 1/f шум [75]



Фигура 78. Музика от кафяв шум [75]

Музиката от фигура 76 (бял шум) е генерирана чрез кодово-графично преобразуване на изходния резултат от функция генератор на случайни числа. Музиката от фигура 78 (кафяв шум) е създадена чрез кодово-графично преобразуване на изходния резултат от първа подред верига на Марков, в която всяка височина има еднаква вероятност да бъде последвана от нейните две непосредствени или предишни височини: за пример, G_4 може да бъде последвано от F_4 , $F\#_4$, $G\#$, A_4 . Това симулира характеристиката на триизмерния спектър на кафявия шум. Музиката от фигура 77 (1/f шум) е генерирана чрез кодово-графично преобразуване на изходния резултат от нелинейното уравнение, предложено от Шрьодер, със стойност за $\mu = 0.5$. Забележимо е че връзките между тоновете драматично наподобява спектралната плътност на кореспондиращите шумове, от които са получени. 1/f шума специално е подходящ за генерирането на последования от музикални събития, чиито съотношения са наполовина между алеаторичните процеси и случайните процеси във веригите на Марков. [26]

4.3.5 ГЕНЕТИЧНИ АЛГОРИТМИ

Преди повече от век и половина, Чарлс Дарвин (*Charles Darwin*) открива, че организмите, които остават неизменни в променящо се обкръжение ще бъдат неспособни да се адаптират към новите обстоятелства или след това ще умрат. Дарвин забелязва, че докато се променят обкръжаващите условия, организмите които по-добре са се адаптирали към новото обкръжение, оцеляват и дават потомство, което наследява

тези характерни черти, докато неадаптиралите се умират. Дарвин нарича този процес “естествена селекция” и го отдава на начина, по който биологичните видове се развиват в природата. Този процес е известен още като “оцеляването на по-силните”. Силата не може да се определи като фиксирано количество. Индивидуалната сила и издръжливост зависи от много фактори: промяна в екосистемата, съревнование между видовете или членовете на дадена популация и т.н. Колкото се променя обкръжаващата среда, толкова това се отразява и на издръжливостта индивидуалните видове. [23]

Подобна аналогия съществува и в алгоритмичната музика.⁶⁹ Генетичните алгоритми в компютъра симулират начина, по който организмите се възпроизвеждат и развиват в природата. Би могло да се каже, че отскоро учените започнаха да симулират биологични процеси чрез компютърен софтуер. Докато природата показва огромна гъвкавост, целта на един компютърен алгоритъм основно е да намери специфичен отговор на специфичен проблем. Един генетичен алгоритъм трябва да осъществява три основни компонента: съревнование, оцеляване и възпроизвеждане.

Една от най-познатите техники за възпроизвеждане в един генетичен алгоритъм е известна като кросоувър (*crossover-кръстосване, пресичане*).⁷⁰ Тази техника комбинира информацията от двамата родители чрез случайно избиране на точка, в която има комбинация на хромозоми. Подобна трансформация наблюдаваме и в алгоритмичната музика.⁷¹ Под внимание взимаме алелите (*alleles-събитията*),⁷² като музикални събития и хромозомите, формирани от алелите, като музикални пространства [28]. Използвайки генетичните алгоритми върху събитийна редица, еволюцията на събитийната редица може да бъде моделирана в съответствие с дефинирани схеми, решени от композитора. [20] Оказва се че генетичните алгоритми са невероятен начин за създаване на редица от вариации в дадена събитийна редица.⁷³

Следващите примери илюстрират как генетичните алгоритми могат да бъдат използвани в музикални композиции. Използва се малка популация, състояща се от 10 индивида (хромозоми) за генерация. Всяка двойка позволява само едно поколение. Хромозомите (събитийните пространства) са ограничени до 4 алели (събития): А, В, С, D. Основния брой от възможни хромозоми е $4^4 = 256$. Тези ограничения са наложени

⁶⁹ Beauchamp, Horner and Haken, “Genetic Algorithms and Their Application to FM Matching Synthesis”, 17-29

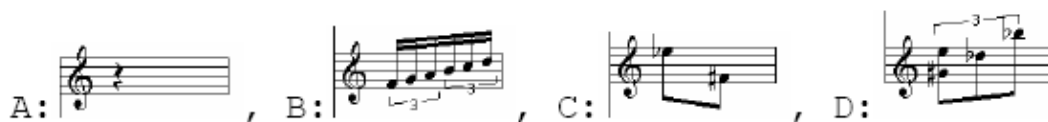
⁷⁰ Виж речника

⁷¹ Goldberg and Horner, “Genetic Algorithms and Computer-Assisted Music Composition”, 479-482

⁷² Виж речника

⁷³ Те успешно биват приложени към звуковия синтез и композиция

поради ефекта на демонстрация, който се цели с примерите. Един наистина функционален пример ще използва по-обширна редица от алели и повече хромозоми. Въпреки това, примерът тук е достатъчен да демонстрира мощта на тази техника (Фиг.79). *Алелите* са дефинирани както следва:



Фигура 79. Генетични алгоритми, генериращи музикални последования [146]

Генетичният алгоритъм в този пример има следните характеристики:

Инициализация: първоначална популация от 10 четворни алели хромозоми, които ще бъдат генерирани произволно. Един възможен хромозом ще бъде например ABBC.

Ще бъде приложена техника за възпроизвеждане с едноточково пресичане.

Вероятността за мутация в поколението е нагласено на 0.10 (1 шанс към 10).

Степента на издръжливост е нагласена според следната схема:

Този генетичен алгоритъм се прилага към събитийната редица от четири allele-хромозоми, създавайка 10 генерации със следните резултати:

ГЕНЕРАЦИЯ 0 (генерирана произволно), Таблица 5

Генерация	Хромозоми
1	ABDB ADAC ACCB BBDB ABDC CDBB CADB BCAC ADAC ABDC
2	BCAC CADB ABDB CADB CDBC CADC ABDC ABDB ABDC ABDC
3	ABDC ABDC ABDC CDBB ABDB CADC ABDC CADB ABDC BCAB
4	ABDC ABDC ABDC ABDB ABDC CADC CADB ABDB CADC CADC
5	CADB CADC ABDC ABDB ABDC ABDB ABDC CADC ABDC ABDC
6	ABDC CADC ABDC ABDC CADC CADB CADC ABDB CADC ABDC
7	CADB ABDB CADC ABDC ABDA ABDC ABDC ABDC CADB ABDC
8	ABDB CADC ABDB ABDC CADC ABDB CADC ABDC ABDB ABDC
9	ABDC ABDB ABDC CADB ADDC ABDB CADC CADC ABDC ABDC

Таблица 5 . Генерация 0

Забележете че поради ограничеността на генетичната информация и популация (само четири алели за хромозом, 10 индивиди за генерация), вариацията в индивидите също е ограничена. Твърде малка генетична вариантност и малка популация причиняват близкородственост, по подобие на истинския живот. Тези примери въпреки това са достъжни, за да покажат как работи един генетичен алгоритъм. Това което следва е музикална транскрипция на генерации 0,4 и 9 (Фиг. 80, 81, 82):



Фигура 80. Генерация 0 [36]



Фигура 81. Генерация 4 [36]



Фигура 82. Генерация 9 [36]

Всеки такт представя един хромозом. Забележете как генерацията се разраства съобразно горепосочената схема за издръжливост. За този пример, хромозоми с голям брой различни алели има по-голям шанс за оцеляване. Например, хромозоми ACAC и BBDB в първоначалната популация нямат голяма вероятност за възпроизвеждане и умират. При генерация 10 популацията доминира чрез хромозом ABDC, който има най-висока степен на издръжливост.

Променяйки схемата на издръжливост се позволява да се създават различни вариации върху популациите. Докато allele A в музикалното пространство е пауза, ще се произвеждат последования, в които тишината ще е доминиращ фактор. Генерираните последования са както следва (Фиг.83):

CAAA DAAB BACB BACB CCDD DAAC BCBV BDDC BCBV CADC

Генерация	Хромозоми
1	BCBD CADB DAAB CAAB CCDC BACB CAAB BDDB CCDB CCDB
2	CAAB CCDB CAAB CCDB CCDB CCDB DAAB BACB BDDB DAAB
3	CCDB CCDB DAAB DAAB CAAB BACB CAAB DAAB CCDB CCDB
4	CAAB DAAB BACB DAAB CAAB CCDB CCDB CAAB BACB CAAB
5	BACB BACA DAAB CAAB DAAB DAAB DAAB CAAB CCDB DAAB
6	CAAB DAAB CAAB DAAB BACA DAAB DAAB DAAB DAAB CAAB
7	DAAB CAAB CAAB DAAB DAAB CAAB BAAB CAAB CAAB CAAD
8	DAAB CAAB CAAB DAAB CAAD CAAB CAAC DAAB CAAC CAAB
9	CAAB CAAC CAAC CAAC CAAB DAAB CAAB CAAD CAAD CAAB

Фигура 83. Генерация 0 (генерирана произволно)

Музикалната транскрипция на генерации 0, 4 и 9 е (Фиг. 84, 85, 86):



Фигура 84. Генерация 0 [36]



Фигура 85. Генерация 4 [36]



Фигура 86. Генерация 9 [36]

Очевидно е как музиката се развива съобразно горепосочената схема. Поради ограничените ресурси, популацията става силно наследствена на генерация 10. Истинската мощ на генетичните алгоритми се проявява най-вече, когато се разполага с голяма популация от хромози, в която генетичната информация е обширна. На музикален език това ще означава по-голямо събитийно пространство с много членове. [28]

4.3.6 ТЕХНИКИ ЗА КОМПОЗИРАНЕ НА АЛГОРИТМИЧНА МУЗИКА

Композирането е артистичен акт, вдъхновен от създаването музика отвъд настоящите технологии или то вече е подчинено неизменно на компютърните технологии.

Развиващите се методи на машинно композиране изискват разрешаването на множество трудни проблеми в областта на алгоритмичния дизайн, представянето на данни и дизайна на човешкия интерфейс, използващ техники от софтуерните технологии, генеративни граматки, компютърни графики. [51]

В Университета в Белкор се прави изследване в тази област, като на няколко слушателя, които са включени в обща телефонна мрежа, се дава възможност да използват няколко програми за композиране и просвирване на музика. В това

изследване се наблюдават 5 алгоритъма за тези програми, които да отговарят на специфичните аспекти на музиката:

“Stochastic Binary Subdivision” - този алгоритъм генерира ритмични структури, които не се повтарят, което избягва някои ритмични проблеми.

“Riffology” генерира допустими импровизации в мелодията, при минимални изчислителни средства.

“Auditals” е граматически (grammar) базирана техника, която набляга на сходствата между структурата на растенията и тази на музиката.

“KeyPhrase Animation”

“Fractal Interpolation” са сходни и са взаимствани от компютърната графика, като се използват за мелодично украсяване.

Подобни алгоритми са наистина полезен инструмент при композирането на музика с компютър.

Изследователите в областта на хуманните науки осъзнаха през последното десетилетие, че композирането и просвирването на музика от машини е едно само по себе си упражнение в проектирането на дизайна на човешкия интерфейс. Системните дизайнери започнаха да разбират, че колкото по-сложни започнаха да стават компютърните системи, толкова самото наблюдение на състоянието на една такава система ще изисква по-високи и по-обхватни нива на комуникация. Звукът се оказва една очевидна възможност. [18]

Да се проектира алгоритъм за композиране на музика, е нещо различно от това да се проектира алгоритъм, който прави инверсия в една матрица [31]. Няма прост, механичен тест за успех.⁷⁴

Други експерти от тази област изказват подобни мнения: «Всеки опит да се симулират възможностите на човека за композиране най-вероятно няма да има успех, докато музикалните модели, които използват хората са описани и моделирани. [57]

Посредством взаимното свързване на два независими проекта в Белкор (Bellcore) (единият изучава приноса на програмирана потребителска телефонна система [65], а другият включва изследване в областта на алгоритмичното музикално композиране), един комплект от програми се проиграва на компютърна система, снабдена с необичайни периферии, които позволяват на компютърна-генерираната музика да бъде достъпна чрез обществена телефонна система. Набирайки (201) 644-2332, слушателите

⁷⁴ Самият Хилър отбелязва “Компютърно-базираната композиция е трудна за определение, трудна за ограничаване и трудна за систематизиране.”

могат да чуят кратка, автоматична музикална демонстрация, състояща се от музикални изпълнения, композирани от компютърни програми [45]. Една от целите при проектирането на музикалното изследване е да се предостави една достъпна музика, базирана на популярни и разбираеми музикални форми (примерно класическата музика на 18-то и 19-то столетие, американска народна музика, съвременна популярна музика) и по този начин да се избегне неяснотата при музикални форми, на които се опира компютърната музика (като електронна и атонална музика).

Това изследване описва пет техники, които се използват от програмите, за да композират музика за автоматизирана телефонна демонстрация. Всеки алгоритъм представя отделна композиционна идея. Комбинацията от тези алгоритми се използва като основа за приложения за различни мрежови трансмии и мониторинг в реално време както и за случайно генериране на музика при филмовата продукция. [38]

“Stochastic Binary Subdivision” е техника, която е проектирана да генерира “trap set” ударни ритми. Новата идея, която стои зад този композиционен алгоритъм е наблюдението, нещо което липсва в доста от алгоритмите за генериране на случайни ноти, понеже тяхната метрична структура е повредена или се игнорира в повечето от случаите. Допълнително наблюдение е това че западната популярна музика обикновено се характеризира в ритмичните си стойности с разделяне на фактор 2 – или т.нар. двуделни ритмични форми, с по-редки случаи на разделяне на 3 или 5, нещо което води до извеждането на много проста идея за генериране на “произволни” ритмични стойности, които винаги се подчиняват на двоичен подразделен модел. Тази идея се доразвива така че да генерира мелодии, с ритмична структура, която следва двоичния подразделен модел.

“Riffology” генерира свободни импровизации, базирани на модели на човешка импровизация. В този модел основната единица на конструкция е т.нар. “riff”⁷⁵, който представлява много кратък мелодичен фрагмент. Последователности от такива “riffs” се събират в процеса на работа на програмата в тр.нар. режим в “полет” (*on the fly*), за да се построят “сола” в една по-глобална рамка на ритмични ударения.

“Auditals” изследва подобността между структури от дадени класове растения с техните вариационни повторения (*repetition-with-variation*) и вариационни повторения, които се срещат в структурата на музиката. Използва контекстно –

⁷⁵ Виж речника

нечувствителни граматика (*context-insensitive grammars*) т.нар. L-система, които за първи път се предлага от Линденмайер [47] за целите на т.нар. нарастващи системи, в които се генерира музика по аналогичен начин на фракталите, предложено като идея от Смит (Smith) [69], за извеждането на графични образи, включващи растения.

“Key Phrase Animation” е техника за мелодично украсяване чрез интерполация между двойки от мелодични сегменти. Аналогична е на техниката, която се използва при графичната анимация, известна като “*key frame animation*”. Плавното визуално движение постигано от аниматорите чрез тази техника, предизвиква интересни противоречия във възприятията, като превежда графиката както в едно плавно, хармонично движение (при бързи темпа), така и в сложно мелодично движение (при по-бавни темпа).

“Fractal Interpolation” е една друга техника за мелодично украсяване. Тя е вдъхновена от работата на Манделброт (*Mandelbrot*) [50], Форниер (*Fournier*), Фусел (*Fussell*) и Карпентър (*Carpenter*) [25] в прилагането на математически или дробни величини за генериране на сцена. Фракталната интерполация използва подобни техники, за да осигури произволно сложно украсяване на музикалните фрази с много прост алгоритъм.

Музикалните генериращи техники описват тук работата си само със синтактични въпроси; програмите нямат методи за разработване на семантики. Семантиката на музиката обхваща едно необятно богатство от сложни взаимоотношения, култура и исторически информация (резултатът от дейностите свободно могат да бъдат наречени “опит”, “възприемане на друга култура”, “обучение”), които все още не съществуват в никоя машинна форма. Оказва се важно и необходимо да имаш опит и да използваш правилна форма, за да се създаде една приятна за слушане музика.

Важно е да се знае, че най-малко две дисциплини изследват и си взаимодействат в проучаването на алгоритмичното композиране – компютърни науки и музика.

4.3.6.1 ПРОГРАМА DDM – МУЗИКА ГЕНЕРИРАНА ОТ АЛГОРИТЪМ “STOCHASTIC BINARY SUBDIVISION”

Метричната структура на западната популярна музика показва една силна тенденция – тя е двоична. Целите тонове се разделят на половини, четвъртини, осмини, шестнайсетини и т.н. Разбира се, срещат се и триделни форми, като триолите, 4 триделни рядко се подразделят в триделни по-кратки стойности, подобно на двуделните. Появява се и ритмично напрежение при тонове, започнали с малки

ритмични стойности да се разширят до по-дълги тонови стойности. За по-голяма прецизност, ако се групират подразделенията на размера в нива, така че n нивото да съдържа всички подразделения, които са нечетни кратни на $2-n$ в размера (например ниво 3 се състои от временни участъци $\{1/8, 3/8, 5/8, 7/8\}$), откриваме, че тоновете започват на ниво n , рядко се разширяват отвъд стойностите на ниво $n-1$, много рядко се разширяват отвъд стойностите на ниво $n-2$ и т.н.

Ритъма, подобно на мелодиите, трябва да поддържа баланса между очакваното и неочакваното. През по-голяма част от мелодията, първоначалното напрежение е заложено на първо време – т.нар. първо силно метрично време с напрежението на второто – относително силно метрично време и така баланса на напрежението се разпределя. Правейки продължителността на тона пропорционална на нивото на подразделение на началния тон, ударението постоянно се връща на първоначалния силен метричен момент. Обикновено опитите да се генерират ритмики на произволен принцип – се провалят, защото те игнорират това изискване. По-усъвършенстваните опити правят специални проверки, за да избегнат разширението отвъд първоначалното или вторичното напрежение, но резултатът е все така непостоянен, защото очакваме да чуем музика, която спазва повторението на най-простите двоични подразделения на размера, на всяко ниво.

```
divvy(ip, lo, hi)
struct instr *ip;
{
    int mid = (lo + hi) >> 1;

    ip->pat[lo] = '|';
    if ((rand() % 100) < ip->density && hi - lo > ip->res) {
        divvy(ip, lo, mid);
        divvy(ip, mid, hi);
    }
}
```

Фигура 87. divvy() [81]

Програмат “ddm”⁷⁶ се опитва да направи музикален ритъм, като се придържа към наблюденията, посочени по-горе и правейки всички останали възможни варианти на произволен принцип. Програмният код “divvy” във фигура 87 е сърцето на програмата ddm. Структурата съдържа **instr**, **density** – по всяка вероятност на всяко

⁷⁶ Акроним за **d**igital **d**rum & **m**elody (или **d**igital **d**rum **m**adness)

ниво ще се появи подразделение на следващо ниво осиг (времевият интервал ще бъде разделен), **res** – най-краткия тон, който може да бъде генериран или най-дълбокото ниво, до което може да се раздели и **pat** - един character string (стринг), в който се съхраняват генерираните образци на тактове. Друга част на програмата ddm се грижи за рутинните задачи за извеждане на входните данни и генериране на MPU изхода.

45:75: 2:0: 96:1 BD

52:75: 4:0: 96:1 SD

57:50: 8:0:120:1 HNC

51:50: 8:0: 64:1 RIM

48:67: 8:0: 80:1 TOM3

54:70: 8:0: 64:1 CLAP

55:75: 8:0: 64:1 COWB

59:50:16:0: 80:1 HNO

53:67:16:0: 72:1 TOM1

Тези данни показват един типичен входен файл за програмата. Първата колона е кода, който трябва да бъде изпратен към drum машината, за да се достигне желан ударен (барабанен) звук; втората колона е плътност (**density**), възможността, която това подразделение ще срещне на всяко ниво (в проценти); третата колона е **res**, най-малкото позволено подразделяне; четвъртата колона е продължителността на звука в шестнадесетворини тонове; петата колона показва колко силен трябва да е звука и шестата колона показва кой MIDI канал (или коя drum машина) трябва да се просвири. Всяка допълнителна информация на линията е коментар, в този случай показва имената на инструментите.

A2	#.....#.....	BD
E3	!.....#.....!.....#.....	SD
A3	!.....!.....	HHC
Eb3	!.....	RIM
C3	!.....#.....!.....!.....#.....!.....	TOM3
Gb3	!.....!.....!.....!.....#.....	CLAP
G3	!.....!.....!.....!.....!.....	COWB
B3	!.....!.....!.....!.....!.....	HNO
F3	!.....!.....#.....!.....#.....!.....#.....!.....#.....	TOM1



Фигура 88. Пример за изходни данни в програмата DDM [81]

Фигура 88 показва изходните данни на програмата ddm в два формата; първият е един такт с информация, показваща резултата от всички разделения, а вторият е два такта от нотния запис на барабаните, показващ крайния резултат. Забележете, че един единствен инструмент се просвирва през цялото време; Знакът ‘!’ индикира разделяне, където даден ударен инструмент иска да бъде просвирен, а знака ‘#’ индикира актуалния ударен инструмент, който се просвирва в това време. Предимство се дава на инструмента, посочен най-рано във входния файл, така че bass drum (BD) просвирва долния такт, въпреки че всички инструменти искат да се появят тогава; по подобие ниският tom-tom (TOM3) се просвирва в средващите 8 тона, имайки предимство пред отворения hi-hat (HNO). Нотния запис на барабаните във фигура 88 започва със същата метрична стойност и след това продължава с друга, която е доста различна, въпреки че се базира на същата редица от вероятности и предимства.

Ако запазим линиите в ddm file описващи тоновете и разликите във височините, вместо типовете барабани, ddm може да се използва за генериране на мелодии. Това е една лека промяна в изпълнението; ударните инструменти вече са кодирани като височинни стойности ($45 \equiv A2 \equiv$ bass drum, $52 \equiv E3 \equiv$ snare drum и т.н.). Единственото допълнение, което е необходимо да се направи е да се кодират височинните разлики. Това се постига като се дефинира всяка стойност от 12 или по-малко, да бъде височинна разлика (по този начин, правим C#0 най-ниския тон, който може да се определи). Добавяйки линии като “1:60:16:31:64:0” и “-1:65:16:31:64:0”, където първоначалната “1” и “-1” означават - “отиди една стъпка по-нагоре” и “отиди една стъпка по-долу”, като могат да бъдат включени нарастващи и намаляващи ходове.

Scale	1,2,4,7,9	
Limits	48,84	
69:33:	8:12:64:0	A4
64:33:	8:12:64:0	E4
1:60:16:28:64:0		up
-1:65:16:28:64:0		down
1:55:32: 4:64:0		up
-1:50:32: 4:64:0		down

Фигура 89. DDM файл за Скат (импровизирана, джазова мелодия) [81]

Фигура 89 показва файл с 8 линии, който се използва за генериране на мелодии, които да бъдат изпяти от Dectalk - синтезатор за говор. Линиите “Scale”(Гама) и “Limits”(Ограничения) принуждават програмата да стои в определена музикална скала (в случая пентатоника) и в определен диапазон (С3, една октава под middle C, до С6, две октави над middle C). Определяйки сравнително кратки тонове в четвъртата колона, вероятността от появата на паузи (необходима за певците да си вземат дъх) нараства.

Фигура 90 е пример за един изходен файл на програмата ddm от файла в предишната фигура. Програмата “scat.c” конвертира изходния MPU формат на ddm в последователност на Dectalk команди, които възпроизвеждат скат пеене (пеене при което импровизирайки, сричките се изпяват в мелодия). Сричките се генерират, като се комбинират случайно избрани хармонични фонemi, със случайно избрани гласни фонemi phoneme; резултатът обикновено е хумористичен.

Един от участниците в това изследване използва ddm два пъти по време на телефонната демонстрация; веднъж да композира една кратка скат последователност за изпяване от посетителя и веднъж да композира откъс за бас, барабани, пиано и клавишет, с названието “Starchastic X”, където X е id (идентификационен номер) на участника – ефективно случайно число. Неформалното тестване на музиката, генерирана с ddm, винаги предизвиква комплименти, без никакви съмнения в нейната музикалност.



Фигура 90. Пример за изходен резултат от Скат пеене [81]

4.3.6.2 RIFFOLOGY (РЕФРЕННА) МУЗИКА, ГЕНЕРИРАНА ОТ “ЕСТЕСТВЕН” МОДЕЛ ЗА ОПТИМИЗИРАНЕ НА СЛУЧАЙНОСТТА

Схеми за използване на случайни събития за композиране на музика датират много години преди компютрите. Измислени са през 18-ти век и се появяват в творбата на Волфганг Амадеус Моцарт “*Musical Dice Game*”, която дава алгоритъм за създаване на музика чрез хвърляне на зарове⁷⁷. Тази схема, базираща се на случайно композиране, е публикувана с провокаторското название (на 4 езика): “Да композираш без никакво музикално познание, толкова много немски валсове колкото ти е угодно, само с хвърляне на дадено число с два зара.” Други добре известни композитори на класическа музика (*Haydn, Bach*) също експериментират с вероятностната композиция [62].

Много след това време, изследователите от Университетите Илинойс и Харвард, Лабораториите на телефонната компания Bell и множество други места, заместват зара с компютрите и променят дисциплини като обработка на сигнали, комбинаторика теория на вероятностите, в посока целите за композиране на музика [30].

Идеята обаче за “riffology”(рефренността) не се появява от тези знамените предшественици. Тя идва от практиката на един водещ китарист в група, изпълняващ импровизационна музика. Една от най-популярните техники за импровизация е била да “съчетаеш заедно много рифове и да ги изспириш наистина много бързо”.⁷⁸ Основното предизвикателство да спириш безкрайни последователности от “riffs” е в това, че не включва никаква гениална мисъл, просто представя добра техника и обширен репертоар от такива рифове. Затова синтаксиса и изпълнението са много важни от семантичното съдържание. Поради тази причина, едно алгоритмично изпълнение на “riffology” (рефренността) е особено подходящо за компютърни програми, понеже те няма да се чувстват възпрепятствани от тяхната неспособност да манипулират семантиката.

Този рефренен алгоритъм дава възможност за избор на динамично-натоварени случайни алтернативи с много параметри - като например кои рифове от дадени мелодични фрагменти да бъдат проспирени, колко бързо да се изспири този риф, колко силно да се изспири, кога да изпусне или да добави тонове, кога да се вмъкне ритмична

⁷⁷ Тази схема открива начини да се подсигури края на последованието, независимо какъв зар е хвърлен.

⁷⁸ Терминът “riff” се използва в смисъл на “постоянно повтаряща се мелодична фраза”, често популярен мотив от известен изпълнител или дадена песен

промяна (break) и т.н. Тези варианти могат да послужат като модел за освободени, но леко лениви и лишени от въображение китаристи.

Основната музикална тема към видео играта “*ballblazer™*”, наречена “*Song of the Grid™*”, е генерирана от този “riffology”(рефренен) подход [46] [44]. Една безкрайно варираща мелодия се генерира от даден мотив/рефрен и се просвирва в акомпанимент на bass линия, барабани и акорди. Самият акомпанимент е монтиран в процеса на работа от репертоар от 4-тактов сегмент, използващ опростена версия на тази “riffology” техника.

Примерната програма “riff.c” (Фиг.91) ще генерира мелодии много подобни на тези в “Song of the Grid”. Акомпаниментът е замислен да има хармоничната структура на подобни мелодии използващи блус скала в A -{A, B, C, D, D#, E, F, G} ще е относително консонатна.

```

/*
**      RIFFO -- Data for Riffology generator
*/
#include <notedefs.h>

#define NPR      8                      /* notes per riff */
#define R        0                      /* rest */
#define H        1                      /* hold */

char      Riffs[] = {
    Eb4, D4, A4, F4, E4, C5, A4, A4,    /* 0 */
    F4, A4, Eb5, D5, E4, A4, C5, A4,    /* 1 */
    Ab4, A4, H, G5, H, Eb5, C5, E5,     /* 2 */
    Ab4, A4, B4, C5, Eb5, E5, Ab5, A5,   /* 3 */
    A4, Bb4, B4, C5, Db5, D5, Eb5, E5,  /* 4 */
    A4, Bb4, B4, C5, E5, Eb5, D5, C5,   /* 5 */
    A4, B4, C5, A4, B4, C5, D5, B4,     /* 6 */

    /* etc. */

    C6, B5, A5, G5, Gb5, E5, Eb5, C5,   /* 39 */
};

int      Numriffs = (sizeof Riffs / (NPR * sizeof (char)));

```

Фигура 91. Riff Репертоар [81]

Инициализацията на “Riffs[]” представлява обема музикално познание на нашия модел на мързелив китарист. Заглавният файл “notedefs.h”, дефинира константи като “A4” и “B4”. Например, “A4” е дефинирана да бъде 0x45 (MIDI “key number” за тона A над middle C), “B4” е дефиниран да бъде 0x47 и т.н. Дефиницията на Riffs[] създава една таблица, съдържаща височинните стойности за един репертоар от 8-тонов мелодичен сегмент.

Всички рифове в този пример са написани в гореспоменатата “блус” A скала. Просвирване в други скали може да бъде постигнато чрез модуляция или чрез

съхраняване информацията за степента на скалата в Riffs[] вместо съхранението на точно определени тонове (въпреки че това ще елиминира възможностите за хроматизъм).

Основната програма, показана във фигура 92 е доста проста; определя броя тактове, който е заявен, засича генератора на случайни числа с текущото време, настройва бързината на изсвирване на стойност шестайсетина ($2^{\text{tempo}} = 2^1$ осминови рифчета за такт, съдържащи шестнайсетини) и въвежда цикъл (loop), за да генерира всеки такт. В този цикъл се решава дали да се промени темпото, с което ще се свири рифа в такта (избирайки сбор 20, 21 или 22 рифове за един такт, това прави осмини, шестнайсетини или трийсет и две вторини), изчислявайки енергията, с която да се свирят рифовете (за целия такт). След което се въвежда вътрешен loop, в който да се избере един riff, който да се изсвири, използвайки pickriff() и накрая да го изсвири. Очевидно това е един прост пример без проверка за грешки, без промени в темпото, освен между тактовете. За да избере следващия за изсвирване риф, програмата изпълнява кода посочен във Фигура 93 много пъти. Всеки път, на случаен принцип избира една възможност (те са от модела, заложен в мисълта на китариста). От всички тях, програмата избира рифа, който е най-лесен и плавен за изсвирване – по-точно рифът, чийто начален тон е най-близкия до една стъпка в скалата встрани, в сравнение с предшестващия краен тон на рифа. Забележете, че поради опита да се избегне ситуация, в която два последователни тона са еднакви, не се правят промени, които са еквивалентни на скок в тритонуса (6 полутонови-стъпки). Една друга важна стъпка в този модел е да се спазва пътя на актуалната пръстовка и позиция на ръката, използвана от китариста. Ако китариста е изсвирил riff 6, завършващ с неговия показалец върху първа струна, 7-та позиция е (B4), C5 ще бъде много по-лесен за изсвирване като следващ тон (среден пръст, първа струна, 8 позиция), отколкото Bb4 (кутре, 2-ра струна, 11-та позиция, изискваща опъване на позицията на ръката, докато 1 струна, 6-та позиция изисква двоен показалец и подвижна позиция на ръката).

Увеличавайки NUMTRIES ще направи преливките между рифовете по-гладки (с повече възможности да се открият перфектни хармонични съчетания), въпреки че колкото по-висок е NUMTRIES, толкова по-детерминиран става алгоритъма.

```

main(argc, argv)
char    *argv[];
{
    int numbars, i, tempo, rpb, dur, energy, r, riff;

    numbars = argc > 1 ? atoi(argv[1]) : 2; /* how many bars? */
    srand((int) time((long *) 0));          /* seed rand() */
    tempo = 1;                               /* initially 2 riffs / bar */
    for (i = 0; i < numbars; i++) {
        if (tempo > rand() % 3)              /* play slower? */
            --tempo;
        else if (tempo < rand() % 3)         /* play faster? */
            tempo++;
        rpb = 1 << tempo;                    /* set riffs per bar */
        dur = BARLEN / (NPR * rpb);          /* calc note duration */
        energy = ecalc(i, numbars);          /* how energetic? */
        for (r = 0; r < rpb; r++) {
            riff = pickriff();                /* pick next riff */
            play(riff, dur, energy);          /* play it */
        }
    }
}

```

Фигура 92. Riffology Main() [81]

```

riff = (rand() % 32767) % Numriffs; /* don't trust rand() */
if (Lstn == 0)                      /* no last note, so */
    return(riff);                   /* anything will do */
dn = abs(Lstn - Riffs[riff * NPR]); /* note distance */
if (dn == 0)                        /* we don't like 0 */
    dn = 6;
if (dn < min) {                      /* save best so far */
    bestr = riff;
    min = dn;
}

```

Фигура 93. Pickriff() Excerpt [81]

```

if (3 * i < numbars)                 /* first third, rests increase */
    return(100 - (90 * i) / numbars);
else if (3 * i > 2 * numbars)        /* last third, decreasing rests */
    return(40 + (90 * i) / numbars);
return(70);                          /* middle third, maximum rests */

```

Фигура 94. ecalc() [81]

Програмата **ecalc()** (Фиг.94) изчислява “енергията”. “Енергията” отговаря на ентузиазма от модела на китариста при свирене на соло; започва силно, отклонява се малко по време на солото и след това достига своя връх отново, когато той/тя осъзнаят, че идва края на солото. Стойността на динамичната енергия, която се връща от **ecalc()** **се използва** да реши дали да изсвири всеки тон от рифа или да пропусне няколко (заменяйки ги с паузи или удължавайки ги с продължителността на предишния тон). Ефектът от това е че солата започват със доста замъглени тонове, стават по-мързеливи и синкопирани в средата и накрая достигат своя връх в края.

В програмата **play()** низът **Biv[]** се използва като начало за “енергията”. Включено е в нея да предразположи програмата да пропуска off-beat тоновете повече, отколкото on-beat. Първоначалният удар основно се пропуска (**Biv[0] = 28**); при положение, че заложената най-ниска стойност за енергия в тази програма е 70, първият тон от всеки риф има поне 98% шанс да бъде изсвирен. Кодът във Фигура 95 се изпълнява веднъж за всеки тон в рифа. Една малка тънкоост в тази програма е че трябва да се буферира всеки тон, докато знаем неговата пълна продължителност, тъй като може да бъде разтеглен с пропускането на следващия тон.

```
int    Biv[]    = {    28, 0, 7, 0, 14, 0, 7, 4,    };

next = Riffs[riff * NPR + i];    /* next note to (maybe) play */
if (next != H && next != R    /* if a normal note */
    && (energy + Biv[i]) < (rand() % 100)) /* but too lazy */
    next = (rand() & 010)? R : H;    /* 50/50 chance rest/hold */
if (next == H) {
    pnd += dur;    /* hold previous note */
    continue;
}
if (pnd)    /* output pending note */
    plink(Lstn = pn, pnd);    /* save last played so far */
pn = next;    /* make this note pending */
pnd = dur;
```

Фигура 95. Play() excerpt [81]

Четириите такта изходен резултат от тази примерна програма (Фиг.96) е типична за такава, която предоставя riffology техники. Необходимо е да се каже, че в тази програма могат да се правят интересни модификации: включване на хармонизирани рифове, проследяване от програмата движението на хармонията така че да се използват ограничени промени в хармонията, позволение за рифове с различна продължителност, позволение за по-сложни ритмични структури в самите рифове, правене на промени в репертоара, в съответствие с позицията в солото и т.н.



Фигура 96. Sample Riffology [81]

4.3.6.3 L-СИСТЕМИ – МУЗИКА, ГЕНЕРИРАНА ОТ *FORMAL GRAMMARS*

През 1960 г. Аристид Линденмайер предлага употребата на паралелни граматични криви (*graph grammars*) за моделиране на нарастване в биологически системи [47]; тези граматика често се наричан “*Lindenmayer-systems*” или просто “*L-systems*”. Алви Рей Смит (*Alvy Ray Smith*) дава описание на L-системите и тяхното приложение в компютърните образи в неговата книга за “графталите” (*graftals*) [69].

Едно предшестващо число пред “L” често се употребява, за да индикира чувствителността в контекста; *0L*-система е контекстно нечувствителна граматика; *1L*-система включва най-близкия съсед в контекста; *2L*-системата включва 2 съседа и т.н. L-система в скоби е тази в която характеристиките в скобите (като ‘[’ и ‘]’, или ‘(’ и ‘)’) се добавят към граматиката като пазачи на място, за да индикират разклоняване, на не са предмет на преместване. Граматиката в L-системите е детерминирана. Това идва да каже, че редици генерирани от L-системните граматика са фиксирани понеже всяко възможно заместване се представя на всяка генерация. Фигура 97 показва компонентите на обикновена заградена в скоби *0L*-система и първите седем низа, които генерира.

Този пример е един от най-простите граматика, които могат да включват два вида разклоняване (например, наляво за ‘(’ и надясно за ‘[’). Името “FIB” е избрано за тази граматика, понеже броя символи (**a**’s и **b**’s) във всяка генерация нараства във фибоначи последователности.

L-системите представят няколко необичайни характеристики и три от тях са представени тук.

Първата е, че те могат да се използват да генерират графични обекти, които приличат на растение. Въпреки че граматиките сами по себе си са доста механични и нямат нито едно от очевидните произволности на природния феномен, графичната интерпретация от генерираните редици са доста убедително “природни”.

Втората характеристика е структурна. Думите от граматиката (получените редици от повторено приложение на заместващи правила) имат фина структура, дефинирана от най-съвременните замени и груба структура, дефинирана от по-ранните замествания. Откакто всички замествания се основават на едни и същи правила, финната и грубата структура са свързани и представят една свободна форма на себеподобие, напомняща структурата на т.нар. “фрактали”.

Третата характеристика на L-системите е една възможност, наречена разширяване на базата от данни (“*database amplification*”), което е умение да се генерират обекти с

внушителна, очевидна сложност от прости редици от правила (голям изходен резултат при малки входни данни).

Генерацията на редиците (или “думите”) в 0L-системите е стриктно дефинирана; всяко възможно заместване трябва да бъде представено на всяка генерация или да представи друг начин, по който всеки символ да бъде заместен. Това означава, че процеса е детерминиран, докато едновременно на лявата страна не се появят две заместващи правила.

В тези случаи където презаписващи правила определят заместването на един символ с много заменими символи, нарастването ще бъде експоненциално.(Фиг.97)

Alphabet:	{a,b}
Axiom:	a
Rules:	a → b
	b → (a) [b]
Generation	String
0	a
1	b
2	(a) [b]
3	(b) [(a) [b]]
4	((a) [b]) [(b) [(a) [b]]]
5	((b) [(a) [b]]) [((a) [b]) [(b) [(a) [b]]]]
6	(((a) [b]) [(b) [(a) [b]]]) [((b) [(a) [b]]) [((a) [b]) [(b) [(a) [b]]]]]

Фигура 97. “FIB”, една проста 0L- система в скоби [81]

Въпреки че генерирането на редиците е добре дефинирано, се изисква една допълнителна интерпретационна стъпка, която да покаже тези редици като графика или като слухови обекти. 192 музикални примери са възпроизведени, опитвайки 12 различни интерпретационни алгоритми на третото генериране на редиците при всяка от 16-те различни граматиките. Примерите варират от 0.0625 такта продължителност (около 0.15 секунди) до 46.75 такта продължителност (около 2 минути). Една малка група оценители (вариращи от музиканти от консерватория до неопитни музиканти) са прослушали в произволен ред всеки музикален пример и са ги оценили по скалата от 0 до 9 ; 0 за “ужасен” през 3 за “почти музикален” и 6 за “приятен”, а 9 за “прекрасен”. От тези 192 примера <89% са оценени над 3⁷⁹. Някои от алгоритмите не успяват да генерират “музикални” примери, но успяват да направят “приятни” такива. Само един от алгоритмите (първият измислен) достига под 3. Ако този алгоритъм се изключи от демонстрацията, успешната стойност ще достигне 95%.

⁷⁹ Средната стойност е била 5.0.

Други изследователи признават музикалния потенциал на тези грамматики. Прусинкиевич описва музикалната употреба на L-system в “*Score Generation with L-Systems*”[63].

Най-простия резултат от общите интерпретационни схеми предоставя един откъс с чертите на фуга, докато различията в граматиката засягат двете мелодии, като ги разклоняват в средата и с малко повече късмет – ги събират в края.

4.3.6.4 KEY PHRASE ANIMATION (В-МЕЖДУ МЕЛОДИЧНИТЕ ЛИНИИ (IN-BETWEENING MELODY LINES))

“*Key Frame Animation*” е техника, която по принцип се използва от карикатурните аниматори, а по-късно се възприема и от компютърните графици. В общоприетата анимация, един аниматор първо рисува кадрите, които съдържат най-важните движения или появата и изчезването на някои обекти. Това се нарича *key frames* – *ключови кадри*..

Веднъж след като ключовите кадри се нарисуват, вътрешните-междинни кадри (“*in-between*”) се възпроизвеждат чрез интерполация между ключовите кадри. Едно от предизвикателствата на този подход е че един главен аниматор може да дефинира по-голямата част от действието и динамиката в една основна последователност, използвайки ключовите кадри и след това да остави останалата работа за възпроизвеждане на вътрешните-междинни кадри на подчинените си.

Ако позволим на мястото на ключовите кадри да се поставят мелодични фрази (съобразявайки изискването, че началните и крайните фрази имат еднакъв брой тонове) и просто се вмъкнат височини и ритмични стойности между съответстващи си тонове във фразите, ще постигнем музикална версия на ключовата кадрова анимация. Взаимствайки тази идея за възпроизвеждането на музикални пасажии, извличаме полза от същата редукция на работа (оставяйки компютъра да свърши по-тежката работа за допълване на фразите), но трябва да се има в предвид, че има все пак различен и учудващ ефект, когато се прилага върху мелодии.

Докато мелодичните фрагменти, които съставляват ключовите фрази са изявени сами по себе си в една временна последователност, то преходите между ключовите фрази са маскирани до известна степен и вниманието ни е раздвоено между слушането на линейни редици от индивидуални тонове и слушането на бавни променливи примери от тонове. Джаз импровизаторите често използват този похват на

трансформация с арпеджо акорди (подобно и на маниера на Бетховен, който може да се чуе в неговата Лунна соната).

Програма “кра”, представя една проста линейна интерполация, използвайки програмата `interp()` да генерира едновременно времеви стойности и динамика във фразата (временните позиции на тона, които се индикират от събитията за натискане (note on) и отпускане (note-off) се отнасят за началото на фразата, колко тежко ще се просвири тона – се определя от динамиката). Линейната интерполация за височинната стойност е ограничена в генерирането на височини в дадената скала – чрез програмата `sinterp()` (използваща ефективен алгоритъм, който открива най-близкия тон в скалата). Това означава, че линейната интерполация е квантазирана по доста неправилен начин. Ако гамата е определена като 12-тонова хроматика (като 0,1,2,3,4,5,6,7,8,9,10,11 аргумент), интерполацията става много по-квантазирана. Фигура 98 съдържа две шестнайсетини нотни фрази с две вътре-измежду техники, генерирани чрез кра, който се вмъква между тях в три стъпки (финалната стъпка генерира новата ключова фраза). Основната гама ({C, D, E, F, G, A, B} “гама по белите клавиши”) е използвана за този пример.



Фигура 98. Пример за ключово-фразова анимация [81]

Понеже височинните стойности на тоновете са ограничени да имат отделни стойности според ограничението на MIDI за представяне на данни, има ограничение за възможното “гладко, спокойно” движение във височина. Освен в случаите когато началните и крайните фрази достигат различни височинни предели или се използва една 12-тонова гама, интерполацията използва много вътрешно-междинни тенденции за възпроизвеждане на сравнително статични резултати. Едно от най-екстремните

движения е представен в настоящия пример, така че четиринадесетата нота във фразата започва от F5 надолу към D4 (една цяла стъпка над middle C), разстояние известно като децима. Повечето от нотите по принцип се местят през терца; така че повече от две или три вътрешно-междинни ще гарантира много повече ноти, отколкото при случайното движение. В някои случаи това може да се създаде много приятен ефект, с бавна промяна на фразата и малко ноти за дадено време. В други случаи, може да бъде по-подходящо да се вмъкне една единствена интерполирана фраза между всяка двойка от оригинални фрази, като по този начин се дублира продължителността на откъса.

В графичната анимация интерполацията рядко е линейна; обикновено при нея всяко последование започва бавно, ускорява се по средата на интерполацията и след това отново се забавя в края, очертаваща S-крива ако позицията се изчертава срещу времето⁸⁰. В такава анимация, обектите, които обикновено се движат са физически обекти, притежаващи маса. Такива внезапни ускорения изглеждат тромави понеже подобно движение ще погълне голямо количество енергия в реалния живот (те произвеждат обширни или непостоянни вторични производни).

В музиката тоновете нямат физическа стойност – нямат маса, въпреки че често се правят асоциации между музиката и физически обекти (полетът на птица, морски вълни и т.н.), така че вероятно някои форми на успокояване и изглаждане на втората производна функция ще се окажат полезно разширение в ключовата фразова анимация.

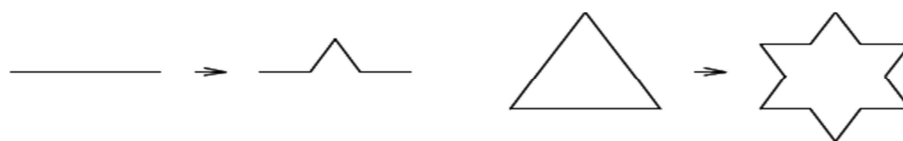
4.3.6.5 ФРАКТАЛНА ИНТЕРПОЛАЦИЯ (*FRACTAL INTERPOLATION*)

През последните десетилетия, артистите от всеки вид изкуство бяха заинтригувани от “естествената произволност” на формите, които се възпроизвеждат по математически път и се наричат фрактали [50]. В частност, композиторите започват да опитват различни начини, чрез които да използват фракталната структура в музиката, като по този начин разработват себеподобността и разширяването на базата данни, които те олицетворяват [24].

Една от най-простите употреби на концепциите, които включват фракталните форми е последователното заместване на права линия с няколко такива сегменти. Снежинката на Кох (*Koch*) е един такъв примерен чертеж.

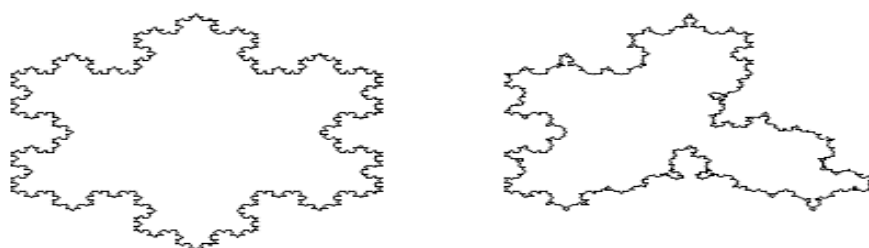
⁸⁰ Това се усложнява в случаи, където движението на два обекта се припокрива, но не започва и не свършва по едно и също време.

В снежинката на Кох всяка една от страните в равностранен триъгълник е заместена от 4 линии, както е показано на фигура 99.



Фигура 99. Генериране на Снежинката на Кох [147]

Ако този процес се извърши безкрайно много пъти, резултатът е една безкрайно дълга линия, която се изпълва до определен предел с произволна плътност. Това позволява да има характеристики някъде между онези едномерни и двумерни обекти. Идеята да имаш дробна величина се отнася към понятието фрактал. Алгоритъмът за Снежинката на Кох е детерминиран - определим (няма възможност за избор; всеки път, когато правите такъв избор той ще бъде абсолютно същия като предишния) и получената безкрайна и сложна крива и видима, методично симетрична. Ако алгоритъма се промени така че да прави някои варианти на случаен принцип (или псевдо-произволно), кривата която ще се получи има много по-естествен вид. Обектът вляво на фигура 100 е Снежинката на Кох след 4 итерации; обектът вдясно е същия, освен че всеки път – на всяка итерация линията се е замествала от 4 линии, посоката на новите издутини – точки по повърхността (навътре или навън) са били избрани произволно и издутините са станали малко по-малки (така че кривата не може да докосне самата себе си).



Фигура 100. Снежинката на Кох след четири итерации [147]

Програмата “fract”, използва една идея, която е презентирана [25] за използване на приблизителни до дробното брауново движение генериране на естествени черти в компютърните образи във филмите.

Идеята е, че позиционирайки добавените линии (когато се замества една линия с няколко, което е описано по-горе) ще се основава на псевдо-случайни числа, които са

получени от позицията на крайните точки в оригиналната линия. Това означава, че въпреки че в получения резултат очевидно липсва симетрията на Снежинката на Кох, изчисленията са възпроизводими. Това е наистина важно при генерирането на филмови кадри, докато една невъзпроизводима схема ще създаде подобни, но различни добавени линии за всеки кадър, причиняваща изкривяване около пейзажа, когато се гледа в редица.

Fract.c интерполира чрез псевдо-произволна схема между нотите в една въвеждаща мелодия. Интерполацията рекурсивно разделя времевия интервал между постигнатите ноти, докато определената резолюция бъде достигната и вмъква нови ноти, избрани в определения интервал. Един параметър за грапавина определя този интервал в условията на връзка между времевия интервал и максималното височинно отместване за всяка нова нота.

С параметър на грапавина (*ruggedness*)⁸¹ настроен на 1, максималното височинно отместване е плюс/минус половин стъпка (полутон) за четвъртина нота.

Максималното височинно изместване е :

$$\pm \text{ruggedness} \times dt$$

където *dt* се измерва в четвъртини ноти. По този начин цялостното височинно изместване се ограничава независимо от подразделената резолюция. За пример, с параметър *ruggedness* със стойност 2, мелодията може да се отклони повече от октава в такт 4/4 .

Подобно на ключовата-фразова анимационна програма, тази програма fract има опция, която принуждава генерираните тонове да се вписват в диатоничната скала (в този случай по белите клавиши {C, D, E, F, G, A, B}).



Фигура 101. Пример за фрактална интерполация [81]

⁸¹ Виж в речника

Примерът в горната фигура (Фиг.101) започва с първия такт (16 удара); след това преминава през: `fract <sample 1.000 -d -r4 -s92062`, за да изведе втория такт (с резолюция 1 четвъртина нота, диатоничен, с параметър за грапавина `ruggedness 4`, произволна стойност отсяваща 92062); и накрая през: `fract <sample 0.500 -d -r4 -s92062`, за да изведе последния такт (с резолюция осмина нота, диатонична, параметър за грапавина `ruggedness 4`, произволна стойност отсяваща 92062).

Като се скицират основните контури за мелодията, която да се следва, определяйки желаната грапавина ‘*ruggedness*’ и определяйки крайната гъстота на нотите, тогава се оставя програмата **fract** да вмъква интересни мелодии.

4.3.7 СИСТЕМАТА *VARIATIONS*

Variations е система за алгоритмично композиране, която се опитва да моделира “трудоемката” страна в творческия процес. Системата е създадена по образец на това как конкретния автор композира музика, което е много подобно на начина, по който се пише канон. Творческият процес при хората позволява да бъдат измислени нови теми винаги, когато това е необходимо или желано. Трудно е да се дефинира творчеството, да не говорим това да бъде извършено от компютърна програма, за това системата не прави опити да създава теми, тя само създава вариации по съществуващ материал. Но при положение, че темата може се получи от вариация върху вариацията, общият резултат е неразличим от създаването на нов открито свързан тематичен материал.[24]

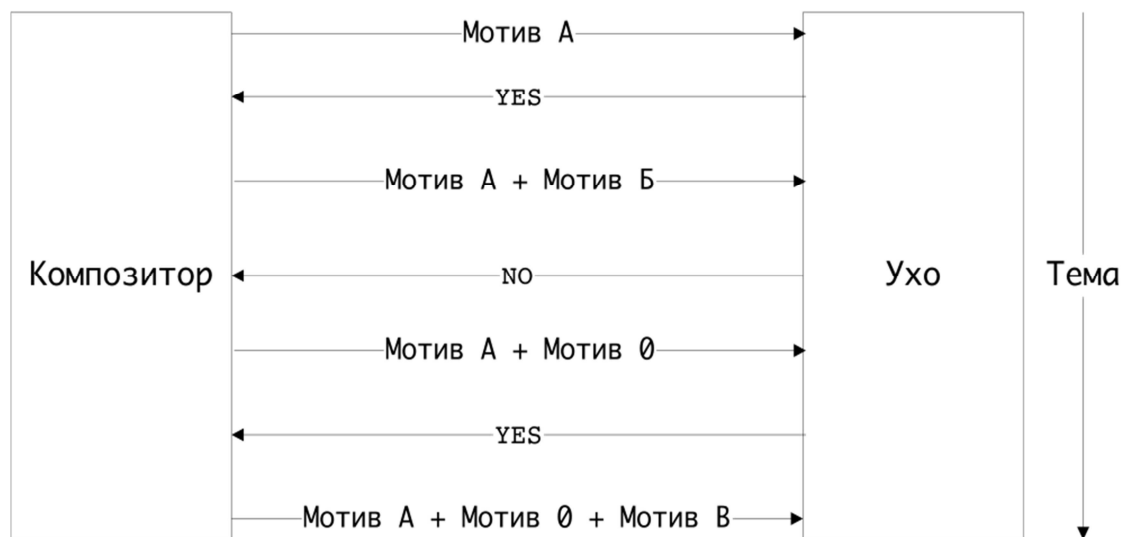
Системата работи на нивото на мотиви, което опростява организацията на музиката. Много по-лесно е да се създаде структурата на дадено произведение, когато се работи на високо ниво, отколкото когато се работи на нивото на свързване на един тон със следващия. Във *variations* редът върху създаваната музика е наложен от набор от правила: да е сигурно, че всички тонове, използвани в получения резултат принадлежат на фрази, свързани една с друга чрез преобразяване. Това позволява на системата да отдава повече внимание на хармоническото развитие.

Ето и самият композиционен алгоритъм:

1. Дефиниране броят на първоначалните теми (мотиви), които ще бъдат използвани в композицията.

2. Композиране на фрази чрез създаване на мотиви. След всяка стъпка се оценява качеството на получената фраза и ако комбинацията е незадоволителна последният мотив се премахва.
3. Създаване на мотив чрез избор на произволен такъв от първоначалните теми и мотиви, вече съществуващи във фразата и варирането на този мотив.
4. При наличието на голямо количество от фрази, те се съединяват в по-големи структури.

Първоначалните елементи на софтуера са модулите, наречени *композитор* и *ухо*. *Композитора* създава материал, а *ухото* го филтрира от всичко “лошо”. Това е моделът производител-потребител, който често се използва в системите за композиране на музика. Едната страна създава музика, другата я консумира и критикува, ефектирайки бъдещата работа на производителя.



Фигура 102. Композитор и ухо в системата variations

На фигура 102 е визуализирана връзката производител-консуматор между *композитора* и *ухото*. *Композитора* взема за входни данни определен брой мелодии и композира фрази от дадена тема мотив по мотив. *Композитора* създава музика като произвежда вариации от предходния мотив и ги напластява и секвентира с другия мотив. Всеки мотив е едновременно копие на първоначалния такъв, негова вариация, копие на предходния мотив или негова вариация. След всяка стъпка резултатът се тества от *ухото* и оценява от него с *да* или *не*. Ако *ухото* “хареса” създаденото процесът на композиране продължава. Ако не го “хареса”, *композитора* изтрива мотива и пробва различна вариация.

4.3.7.1. МОТИВНО ОРИЕНТИРАНАТА ОРГАНИЗАЦИЯ В КОМПЮТЪРА

Системата организира и манипулира материала на нивото на мотивите, така че тематичното развитие е присъщо на създаваната музика. Мотивно ориентираното вътрешно представяне налага подразбираща се структури в процеса на композиране; тя позволява лесното и тясно манипулиране на материала, така че получените тематични вариации се нито тривиални, нито неразпознаваеми.

Всеки мотив съдържа следното:

- Списък от тонове, където възможните стойности са 1 (тоника), 2, 3, 4 (субдоминанта), 5 (доминанта) и т.н.
- Съответващ списък за ритъм и артикулация
- Информация за тоналността: тониката (включително и от коя октава е), дали ладът е мажорен или минорен
- Информация за времетраенето: чрез такт от предишен мотив
- Информация за тембъра

Избраната форма поощрява трансформациите: модулацията е просто средство за промяна на тониката или добавяне на номер на всеки член от списъка с тоновете. Промяната в метрума е също толкова проста: осъществява се чрез разделяне на елементите на метрума, като например деленето на 2 превръща всяка четвъртина нота в осмина, всяка осмина в шестнайсетина и т.н. По същия начин променяйки номерата в артикулацията прави легатото да звучи по стакато.

Тъй като системата първоначално започва с малък брой определени фрази, цялата композиция се състои от мотиви, тясно свързани помежду си. Всяка фраза съдържа себеподобни компоненти и всички фрази създадени върху едни и същи мотиви са свързани по между си. Съдържащите се мотиви са свързани чрез трансформация и като връзките могат да бъдат съвсем очевидни или почти неувимими.

4.3.7.2 ПРОЦЕСЪТ НА АЛГОРИТМИЧНОТО КОМПОЗИРАНЕ

Композирането по същество е тежка творческа работа и по същество е алгоритмично. Описаният по-горе начин на композиране не е присъщ единствено на компютъра, той е доста сходен с процеса на композиране при хората. Фразите са построени от мотиви един по един. Мотивите са взети по избор от няколко теми. За пример ще започнем със следната известна тема **от соната на Моцарт**, (Фиг.103):



Фигура 103. Сонатна тема на Моцарт [148]

Ако дължината на последвалата вариация е седем ноти, тогава следващото последование може да е възможния резултат (Фиг.104):



Фигура 104. Възможен резултат [148]

Следнотопо следование може да бъде също толкова приложимо (Фиг.105):



Фигура 105. Още една вариация [148]

Ако дължината трябва да бъде по-къса или по-дълга от седем ноти, може се изведе проста вариация като тази (Фиг.106):



Фигура 106. По-къса вариация [148]

Могат да се видят и още по-драматични промени (Фиг.107):



Фигура 107. Още по-къса вариация [148]

Може да се промени тоналността (Фиг.108):



Фигура 108. Промяна на тоналността [148]

Може да се умножат стойностите на височините по 1,5 и да се закръглят (Фиг.109):



Фигура 109. Умножаване на стойностите [148]

Може се да умножат стойностите на височините по 1,5 и да се отреже неточната част от резултата (Фиг.110):



Фигура 110. Отрязва част от резултата [148]

Може да се умножат стойностите на височините $\times 2$ (Фиг.111):



Фигура 111. Стойностите умножени $\times 2$ [148]

Когато започнат да се комбинират промените една с друга се получава комплексно поведение. Вариациите все по-малко започват да приличат на оригиналната тема и все повече на претеглените вероятностни характеристики на процесите на Марков приложени при избора на подходящи следващи тонове, ритъм и артикулация. Въпреки това все още е ясно, че съществува връзка между мотивите и произлязлото от тях.

За пример може да се умножат височините $\times 3$ и да се промени тоналността (Фиг.112):



Фигура 112. Стойностите умножени $\times 3$ [148]

Можем да се избере по-кратък участък, да се умножи стойността на ритъма $\times 2$, да се изберат различни подмножества на височините и ритъма и да се транспонира (Фиг.113):



Фигура 113. Умножени ритмични стойности $\times 2$ [148]

Може да се избере и по-дълъг участък, който да се върне в до мажор и да се размести ритъма в него (Фиг.114):



Фигура 114. По-дълъг участък [148]

Всичко става още по-интересно, когато се варират самите вариации. Всяка вариация на тема е като фотокопие или музикален запис с лошо качество – с всяка итерация приликата с оригинала става все по-малка. След няколко итерации оригиналът е почти неузнаваем. Ако вземем под внимание последните три примера, които са доста разпознаваеми като за начало, накрая се получават мотиви които едва наподобяват оригиналната тема.

Например може се да промени първата тема като се разделят стойностите на височините на 4, да се избере различно подмножество от ритмични стойности и да се транспонира (Фиг.115):



Фигура 115. Разделяне на ритмичните стойности на 4 [148]

Всички тези модификации са класически вариации, използвани в каноните и фугите, за да варират тематичния материал. Няма нищо особено или вдъхновено в примерите, те са създадени, за да демонстрират съществуващият изключително голям брой на възможните вариации върху дадена пиеса. А те се много повече, от тези които един човек би поискал или би могъл да проучи. Красотата в използването на компютър е че той *може* да проучи космоса.

4.3.8 ЦЕЛТА НА АЛГОРИТМИЧНОТО КОМПОЗИРАНЕ

Има два различаващи се типа на творчество – вдъхновението и процеса на усилна работа. Алгоритмичното композиране е старо, колкото музикалната композиция. Често се приема за средство, чрез което композиторът улеснява работата си и влага по-малко талант. Може да бъде прието като композиционно средство, което помага на композитора да работи по-бързо. И в двата случая възприемаме алгоритмичното композиране, като средство и начин. Изпълнението на компютърния алгоритъм може да бъде конкретизирано до два компонента - как да се разбере един творчески процес, който да се пресъздаде чрез алгоритъм и как да се програмира компютъра да различава хубава и лоша музика. Въпросът какво е алгоритмично композиране е аналогичен на въпроса какво е изкуствен интелект. И двете понятия са дебатираны горещо. Няма еднозначен отговор. [22] [30]

Алгоритмичното композиране е приложението на устойчив, строго-дефиниран алгоритъм към процеса на композиране на музика. Възприема се неодобрително от

традиционните композитори, защото често се използва като значение за увеличение на музикалните образци. Истината е че музика, продуцирана от алгоритмичен алгоритъм се счита за по-нисше стояща, не защото е създадена от алгоритъм, а защото е ничия музика – принадлежи по-точно на дизайнера на алгоритъма – не на потребителя на алгоритъма. Един творец може да избегне този негативизъм, като създаде собствен алгоритъм и музиката му принадлежи изцяло. Единствената разлика между творческата методология на композитора и един алгоритъм е тази, че композиторът може да изрази и представи повече личен поглед и усет. Един алгоритъм по принцип е неизменен, докато творчеството често разбива правилата. Но трябва да се каже нещо за следването на правилата чрез изпълнението им. Това е което алгоритъмът прави доста лесно, но което един композитор доста често избягва, понеже ние искаме да разбиваме правилата в името на творческия патент. Понеже е трудно да се следват правилата от един творец, има причина защо строго-дефинирани структури се използват за създаване на музика, изкуство, скулптури, танци и поезия: за да следва правилата, един творец винаги ще измисли творческо решение, което иначе не би било избрано или измислено. Именно поради тази причина, компютърът се вписва добре в задачата за креативност чрез усилна работа, като е неспособен да излезе дори на крачка от строго дефинираните структури.

За да се върнем към изкуствения интелект, ако един алгоритъм вярно представи творческия процес на един артист, каква тогава е разликата между музиката, създадена от един творец и музиката, създадена от един алгоритъм?

Творчеството най-вече е процес на спонтанност, когато се дава път на вдъхновението, усещането, таланта. В други случаи някои творци прекарват дълги часове в обмисляне и преработка, като композирането им струва много усилия и чувство на безсилие. С две думи – творчество идва по два начина – чрез талант или чрез много работа. В първия случай може да се създаде много вдъхновена и красива музика, която обаче другите да не разберат. Докато втория случай наподобява на итеративен алгоритъм, който се опитва да достигне някои оптимални функции, с добри качества и следователно е по-лесно осъществимо чрез една компютърна програма. Заради това дефинираме ясно целта на една алгоритмично композиционна система: да пресъздаде творческата методология на композитора, когато композиторът създава музика не на базата на вдъхновението си, а на упорита работа. Резултатът е пригодена система за този специфичен тип композитор; ако друг творец е използвал системата, за

да създаде музика, тогава той композира музиката на първия композитор, не своя собствена.

От философска гледна точка когато говорим за разликата между музикалното композиране и музикалното възприемане – става въпрос за интересно отклонение. Когато един алгоритъм, написан от композитор, създава музика, която не е точно тази, която композиторият желае, той може да я филтрира - той преработва частите, които не му харесват. Удачно е да се зададат въпроси, като колко минимални трябва да са промените в алгоритмично създадената музика, за да бъде сметната за композиране и композирана от алгоритъма ли е тя или от композитора? Да предположим, че алгоритъмът е машина, измислена от композитор, която натиска клавишите на пианото повече или по-малко на случаен принцип. Композиторият записва тази клавишна сесия, за да я редактира по-късно и съчетава няколко такива сесии, докато получи една добре звучаща композиция (по негова преценка). Каква е ролята на композитора в този случай? Дали композиторият композира музика или само разпознава музиката?

Друг пример е когато композиторият използва машина за създаване на музика, чрез която той записва сесията, редактира я и довежда до краен продукт, който нарича „Рапсодия”. Заслугата за създаването на тази композиция е на машината; на дизайнера на машината; на потребителя на машината, който разпознава тези пасажии, като осъществими за композиране или тази Рапсодия не е музика, звуците на която не са музикални, защото всъщност не са били преживянии и чути от твореца, въпреки че тази дефиниция категорично не дава обяснение на този процес?

Една гледна точка е тази, че изкуството е връзка и комуникация и всяко едно нещо, което си комуникира – е изкуство. Самият акт на съзерцаване – е вид комуникация и връзка между човека и обекта на съзерцаване, независимо, че след това артистият ще се прибере у дома и ще пресъздаде този момент в творчеството си – или както е прието да се казва – ще подражава на природата. [23]

Друга гледна точка е и тази, че само това, което се получава директно от ръцете на артиста, принадлежи на артиста. Докато първата гледна точка счита визията и вдъхновението за много по-важни от самия акт на труд и създаване, втората гледна точка счита труда за много по-важен от самото вдъхновение. Първата гледна точка напълно обезценява алгоритмичното композиране, докато другата напълно го осмисля и обхваща. Тук няма отговори, това е просто начин да се предположи, че един алгоритъм много по-тясно рефлектира върху методологията на композитора, като по-незначителния въпрос е дали работата е автентична и принадлежи на композитора.

ПРИНОСИ

Всички гореизброени методи са описани с цел намирането на най-оптималната система в наличните лабораторни условия, за да бъде създаден алгоритъм на музика. Четвъртата глава от дисертацията, озаглавена “Методи за създаване на алгоритмична музика” е изцяло насочена към областта на изследване, свързана с алгоритмичната музика и методите за нейното създаване. Това изследване включва най-известните методи, които се използват за алгоритмично композиране - стохастични процеси, вероятностни функции, веригите на Марков; дву и триизмерни хаотични системи; фрактали; алгоритми получени от спектрален шум; клетъчна автоматизация, генетични алгоритми, L-системи. Подробно са дадени примери за употребата на L-системите в композирането на алгоритмична музика, като става ясно, че статистическата себеподобност играе важна роля в музикалната интерпретация на L-системите, защото доставя повече вариации и прави музиката по-малко монотонна и предсказуема. L-системите са едно необятно поле за музикални експерименти.

Важна част от тази глава е и частта свързана с клетъчната автоматизация, като са изведени конкретни примери за имплементацията на СА към музикалната композиция. Основна част от тази глава на дисертацията е тази свързана с фракталите и създаването на фрактална музика. Изследват се методи за създаването на фрактална музика, както и специфични черти, обуславящи създаването на тази музика. Освен гореспоменатите методи за композиране, в тази глава е засегната и употребата на веригите на Марков, като източник на алгоритмично композиране, а отделно са представени и основни кодово-преобразуващи техники в музиката. Разглеждат се атрактори от дву и триизмерни хаотични карти, употребата на генетични алгоритми, както и генерирането на музика от шум.

В настоящата глава се дава сравнително цялостно описание на пет алгоритмични музикално-композиционни техники. Техниката *Stochastic binary subdivision* може да се използва за генериране на партии на барабани, мелодии или дори на цяла оркестрация. *Riffology* техниката генерира музикално мелодични движения. L-системните граматики генерират редица, която може да бъде интерпретирана за генериране на музикални звуци. *Key phrase animation* може да се използва да генерира движещи се мелодични линии от статични мелодични извадки. *Fractal interpolation* може да добави украшение

към по-сеплите мелодични линии. Интересно е да се отбележи, че много от музикалните фрагменти, генерирани от тези алгоритми имат семантично/емоционално съдържание, (някои изглеждат мрачни, други са щастливи и енергични). Докато алгоритмите сами по себе си нямат информация относно човешките емоции, трябва да заключа, че всяка семантично значителна конструкция, която се среща е съвпадение.

Нашата семантична интерпретация на тези случайни събития затвърждава способността ни да намираме смисъл във всичко, което представлява предизвикателство за нашия интелект и креативност.

В настоящата глава обстойно се разглежда и системата за алгоритмично композиране, наречена Variations, като прототип на конкретния автор, който композира музика. Като обобщаваща част от тази глава е частта, в която се изследва целта на алгоритмичното композиране, като от една страна целта му е вдъхновение за композитора и от друга страна е средство за постигане на един композиционен процес.

ГЛАВА 5. СРЕДСТВА ЗА СЪЗДАВАНЕ НА КОМПЮТЪРНА АЛГОРИТМИЧНА МУЗИКА

Това което ще бъде представено в тази глава от дисертацията са средствата, които се използват за създаването на компютърна алгоритмична музика. С оглед на практическата реализация свързана с композирането на фрактали е необходимо да се направи сравнителен анализ между известните за целта софтуерни продукти и да се подбере най-подходящия. В повечето случаи тези софтуерни продукти имплементират повечето от алгоритмичните процедури, които бяха обсъдени в предишната глава. Едно от най-големите предизвикателства на тези програми е че записват пиесите, които създават в MIDI формат, което позволява по-нататъшното обработване на тези пиеси в допълнителни софтуерни програми. MIDI както разбрахме от предишните глави е съкращение за *Musical Instrument Digital Interface*, което представлява комуникационен протокол, който позволява на инструментите и секвенсерите (както и на компютърните музикални програми) да си комуникират помежду си, просвирвайки, редактирайки и съхранявайки музика [51]. Всеки MIDI интерфейс има 16 канала, към които се насочва цялата MIDI информация (височина, трайност, динамика, инструмент). Всеки канал е полифоничен, като броя на гласовете за всеки трак зависи от качеството на вградения MIDI борд. MIDI файла е компютърен файлов формат, който съхранява MIDI информацията (височина, темпо, трайност, динамика) по много компактен начин. Тези файлове могат да бъдат заредени в музикални нотиращи или секвенсерни програми, за допълнителна редакция и обработка на музикалния материал. Ето и най-известните програмни продукти, които могат да се използват за създаване на компютърна алгоритмична музика.

FRACTMUS (ФРАКТАЛИ)

Автор: Густаво Диаз Жерез (*Gustavo Diaz Jerez*)

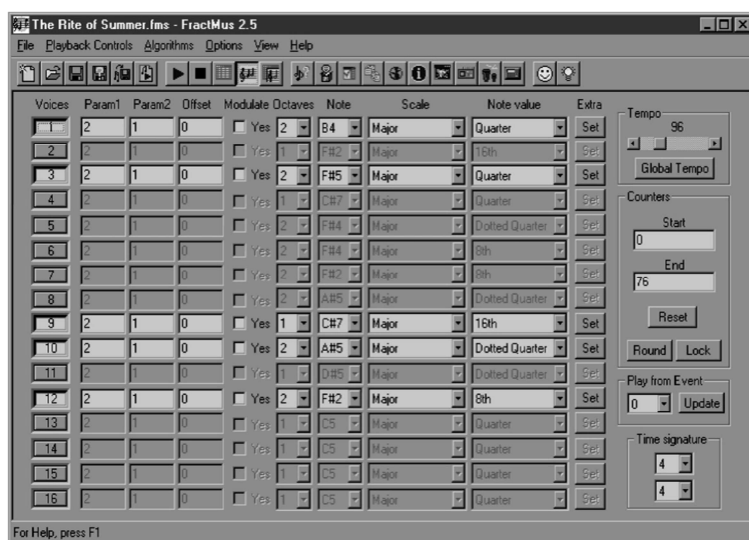
Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: (<http://www.gustavodiazjerez.com/fractmus>); <http://www.musoft-bulders.com>

FractMus е програма, която генерира музика от итеративни алгоритми, създадена от испанския пианист и композитор Густаво Дياز Жерез (*Gustavo Diaz Jerez*). На разположение в програмата са 12 итеративни алгоритми от хаотичните динамични системи, разгледани подробно в Глава 5 от настоящата дисертация: редицата на *Morse-Thue*, *Logistic Map*, *1/f Noise*, *Henon attractor*, *Hopalong attractor*, *Martin attractor*, *Gingerbread man attractor*, *Lorenz attractor*, *3n+1 numbers*, *1D Cellular Automata*, *noise (1/f noise)*. Програмата също така включва линеен генератор на случайни числа, който е полезен при стохастичните процеси. Алгоритмите могат да бъдат присъединени независимо към всички MIDI гласове. Всички музикални параметри могат да бъдат контролирани от потребителя (тоналности, трайност, динамика, инструменти), което позволява широка гама от кодово-графично преобразуване. В програмата могат да бъдат дефинирани независими секции (събития), всяка от които да притежава свой собствен набор от алгоритми, музикални параметри и кодово-преобразуващи техники.



Фигура 116. Основен прозорец на програмата FractMus [141]

Тази програма е много приятна за употреба: разполага с 16 гласа, на които композитърът може да определи различен алгоритъм (Фиг.116). Като допълнение, композитърът може да определи параметри като музикална тоналност, първоначален тон и основна ритмична фигура за всеки един от гласовете. Алгоритмите са повтарят съобразно брояч, който наблюдава ритъма на различните гласове и на всяка итерация се подбира нов тон от избраната тоналност. Програмата е изключително добро средство за създаване на учудващо приятна музика.

FractMus превежда композициите в графичен образ. Основно програмата взима всички композиционни параметри (тоналност, трайност, гласове, алгоритми) и от тях

създава графичен образец, подобен на мозайка, който има фрактална структура. Всеки такъв образец се преобразува уникално към всяка композиция. Програмата също така предлага 15 дефинирани разновидности на мажора и минора, както и различни ладове (Фиг.117), от които да изберете тези, които ще използвате в създаването на дадена композиция.

	Натурален Мажор
	Натурален Минор
	Хармоничен минор
	Мелодичен минор
	Лидийски лад
	Миксолидийски лад
	Дорийски Лад
	Фригийски лад
	Локриански лад
	Акустичен лад
	Октатоника 1
	Октатоника 2
	Пентатоника
	Целотонна гама
	Хроматична гама

Фигура 117. Дефинирани разновидности на мажор и минор [141]

Програмата предоставя 11 дефинирани тонови трайности: точкувана цяла нота, цяла нота, точкувана половина, половина, точкувана четвъртина, четвъртина, точкувана осмина, осмина, точкувана шестнадесетина, шестнадесетина, трийсет и две вторина.

Също така FractMus предлага възможност за инверсия (Фиг.118) обръщайки наобратно една мелодия, което е композиционна процедура, в която посоката на интервалите в мелодията се обръща - квинта нагоре става квинта надолу, секунда надолу става секунда нагоре. За улеснение програмата обръща от С (До), така че взима тона С (До) като опорен тон за обръщението. Например:



Тази мелодия е в ла минор и нека сега ѝ приложим обръщение:



Фигура 118. Инверсия [141]

Както виждате всеки тон от оригиналната мелодия е запазен, но отива в противоположната посока. Понеже програмата взима тона С (До) като опорен за обръщението, има кодово-преобразуваща схема за всяко обръщение:

<u>Оригинален тон</u>	<u>След обръщение</u>
С	не се променя
D-flat	В
D бемол	В-бемол
Е-бемол	А
Е	А-бемол
F	G
G-бемол	не се променя
G	F
А-бемол	Е
А	Е-бемол
В-бемол	D

B

D-бемол

След като се обърне една мелодия, нейната оригинална тоналност се променя, но в програмата е възможно да се запази оригиналната тоналност чрез функцията *Match Scale*.

Чрез програмата FractMus съм генерирала следните музикални примери, които могат да бъдат чути в *Приложение 4, папка FractMus*:

1. Morse Thue (Приложение 4, Morse Thue.mid)

Voice 1: Algorithm: Morse-Thue (2,3)

Octaves: 2

Scale: Major

Starting Note: C3 (MIDI note # 36)

Instrument: String Ensemble

Voice 2: Algorithm: Morse-Thue (2,6)

Octaves: 1

Scale: Major

Starting Note: C5 (MIDI note # 60)

Instrument: Grand piano

Voice 3: Algorithm: Morse-Thue (2,4)

Octaves: 1

Scale: Major

Starting Note: C5 (MIDI note # 60)

Instrument: Music Box

2. Wolfram Cellular Automata (Приложение 4, Wolfram.mid)

Voice 1: Algorithm: Wolfram (Rule 25, Size 128)

Octaves: 1

Starting note (B3 MIDI note #47)

Scale: Mixolydian

Instrument: Chorus piano

Voice 2: Algorithm: Wolfram (Rule 30, Size 128)

Octaves: 1

Starting note (C5 MIDI note #60)

Scale: Mixolydian

Instrument: Viola

3. $3n + 1$ numbers (Приложение 4, 3n+1_1.mid; 3n+1_2.mid)

3.1. Voice 1: Algorithm: $3n + 1$ numbers (33)

Scale: Minor Melodic

3.2. Voice 1: Algorithm: $3n + 1$ numbers (123)

Scale: Lydian

4. Logistic Map (Приложение 4, Log Map1.mid; Log Map2.mid)

4.1. Voice 1: Algorithm: Log Map (3.9)

Scale: Major

4.2. Voice 2: Algorithm: Log Map (2.4)

Scale: Major

5. $1/f$ noise (Приложение 4, 1f noise_1.mid; 1f noise_2.mid; 1f noise_3.mid)

5.1. Voice 1: Algorithm $1/f$ Noise (0.45)

Scale: Dorian

5.2. Voice 2: Algorithm $1/f$ Noise (1)

Scale: Dorian

5.3. Voice 3 : Algorithm $1/f$ Noise (0.2)

Scale : Dorian

6. Henon (Приложение 4, Henon1.mid; Henon2.mid)

6.1. Voice 1 : Algorithm : Henon (1.7, 0.4)

Scale : Minor natural

6.2. Voice 1 : Algorithm : Henon (2, 0.3)

Scale Minor natural

7. Martin (Приложение 4, Martin1.mid; Martin2.mid)

7.1. Voice 1 : Algorithm : Martin (3.618)

Scale : Major

7.2. Voice 2 : Algorithm : Martin (3.141590)

Scale: Major

8. Gingerbread (Приложение 4, Gingerbread1.mid; Gingerbread2.mid)

8.1. Voice 1: Algorithm : Gingerbread (-0.1, 0.3)

Scale: Pentatonic

8.2. Voice 2: Algorithm : Gingerbread (-1.1, 0.8)

Scale: Pentatonic

9. Lorenz (Приложение 4, Lorenz1.mid; Lorenz2.mid)

9.1. Voice 1: Algorithm : Lorenz (0.9)

Scale: Major

9.2. Voice 1: Algorithm : Lorenz (0.3)

Scale: Major

10. Random (Приложение 4, Random1.mid; Random2.mid)

10.1. Voice 1: Algorithm: Random (0, 0.9)

Scale: Major

10.2. Voice 1 : Algorithm : Random (0.3, 0.6)

Scale: Major

11. MIX Composition (Приложение 4, MIX.mid)

Voice 1: Algorithm : Wolfram (Rule 30)

Instrument: Grand piano

Scale: Mixolydian

Voice 2 : Algorithm : Morse-Thue (2,1)

Instrument: String ensemble

Scale: Mixolydian

Voice 3: Algorithm: Log Map (3.9)

Instrument: Bright piano

Scale: Mixolydian

Voice 4: Algorithm : Gingerbread (-0.1, 0.5)

Instrument: Pan flute

Scale: Mixolydian

FRACTAL COMPOSER

Shareware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: www.fractalcomposer.com

Приятна и лесна за генериране на фрактална музика е програмата Fractal Composer. Тя създава музика с помощта на принципа на самоподобие. Обикновено започва с кратък мотив като част от музикална фраза, която по-късно се развива. Програмата композира различни произведения в различни стилове, различна скала, тоналности, темпо и размер, дава възможност да бъдат подбрани и инструментите.

Когато композираме произведение, кликваме първо върху бутона Compose и от падащото меню избираме скалата, в която ще композираме, например Chromatic Scale . След това избираме желанния размер, да кажем 4/4 и темпо, например 90. Записваме нотите, като използваме буквените им означения, октавова принадлежност и нотни стойности. Ако искаме да прослушахме направената до този момент мелодия, кликваме върху нотите, намиращи се вдясно от полето за писане на ноти. След като създадем мелодията, можем да изберем и различните инструменти, с които да звучат отделните гласове, тъй като Fractal Composer може да генерира многогласно нашата мелодия. Това става от меню Voices, като от падащите менюта - Voices 1, Voices 2, и Voices 3 избираме инструментите за всеки глас. След като сме композирали мелодията и сме сложили различните инструменти, можем да чуем какво сме направили, като кликнем върху бутона Generate Piece. Получената мелодия може да бъде записана в компютъра (това става в зависимост от програмите и драйверите, инсталирани на компютъра - има доста начини и програми за това). Композирането с тази програма е лесно и забавно,

има много варианти, които може да се изпробват при подбор на мелодия и инструменти.

ART SONG AND MUSICLAB I - MUSIC FROM CHAOS

Автор: Дейвид Стробийн (David Strohbeen)

Shareware

Платформа: Windows

Поддръжане на MIDI: Да

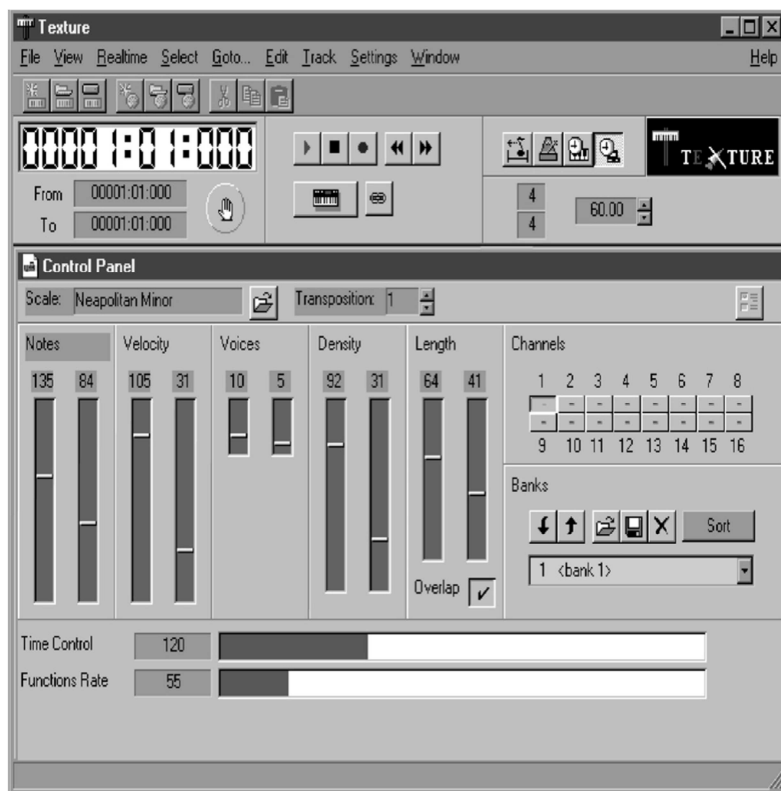
Наличност: <http://www.fractalmusiclab.com;>

<http://www.members.aol.com/strohbeen/fmlsw.html>

Странни атрактори (*Strange attractors*) от хаотични динамични системи, използващи итеративни функционални системи (фрактали) и уравнения от втора степен, като образи предоставени от потребителя. Тези програми, създадени от Дейвид Стробийн (*David Strohbeen*), използват итеративни и квадратни уравнения, за да създават странни атрактори, от които да се получава и самата музика. Те също така могат да трансформират графичните образи в музика. Голям брой от музикални параметри са настроени за потребителите, позволявайки да се създават различни музикални стилове. Потребителят има контрол над всички музикални параметри, като трайност, обхват на тоновете и т.н. Параметрите могат да бъдат променени в реално време. Всяка една от тези програми генерира стандартни MIDI файлове.

TEXTURE

TEXTURE е програма за композиране на алеаторна музика (музика включваща елементи, избрани на случаен принцип), създадена в Аржентина от композитора Луис Мария Рохас (*Luis Maria Rojas*). Тази програма е алеторен генератор на MIDI събития, чийто резултати могат да бъдат съхранявани, записвани, редактирани и просвирвани чрез вграден секвенсер. Може да се използва и в реално време. Програмата има Control Panel, който се състои от плъзгачи за спецификациите на максималните и минимални стойности, които могат да бъдат генерирани за различни аспекти в песента, като тонове, динамика, брой гласове (Фиг.119). Тези плъзгачи могат да бъдат програмирани да се променят в реално време, съобразно функциите, които са изрисувани ръчно или създадени от приложение за генериране на математически криви.



Фигура 119. Основен прозорец на програма Texture [149]

MUSINUM (ЧИСЛОВА ТЕОРИЯ)

Автор: Ларс Киндерман (Lars Kindermann)

Платформа: Windows

Freeware

Поддържане на MIDI: Да

Наличност: www.forwiss.uni-erlangen.de/~kinderma/musinum; <http://reglos.de/musinum>

MusiNum – е създадена от Ларс Киндерман (*Lars Kindermann*) и е програма, която генерира MIDI тонове от числовата редица Морз-Тю (*Morse-Thue*). Музикалната машина използва брояч, чрез който потребителят може да определи различни модели на броячи и базисна система за числата. На всяка стъпка броячът произвежда MIDI тонове. За да се разбере как програмата създава тонове, нека да вземем за пример един брояч, който брои двоични числа в единични стъпки: 0, 1, 10, 11, 100 и т.н. В този случай тоновете се изчисляват чрез добавяне на цифрите, които са еднакви с 1 във всеки един от тези редици от двоични числа. Ако резултатът е равен на 1, тогава това съответства на тона С (До); ако резултатът е равен на 2, това съответства на тона D (Ре); ако резултатът е равен на 3, това съответства на тона Е (Ми) и т.н.(Фиг.120). Могат да бъдат получени различни тонови поредици, чрез различни стъпкови модели и

базисни системи, чрез прозореца Voice Edit (Фиг.121). Също така могат да бъдат избрани различни настройки за различните гласове, за да се генерира полифоничен материал.

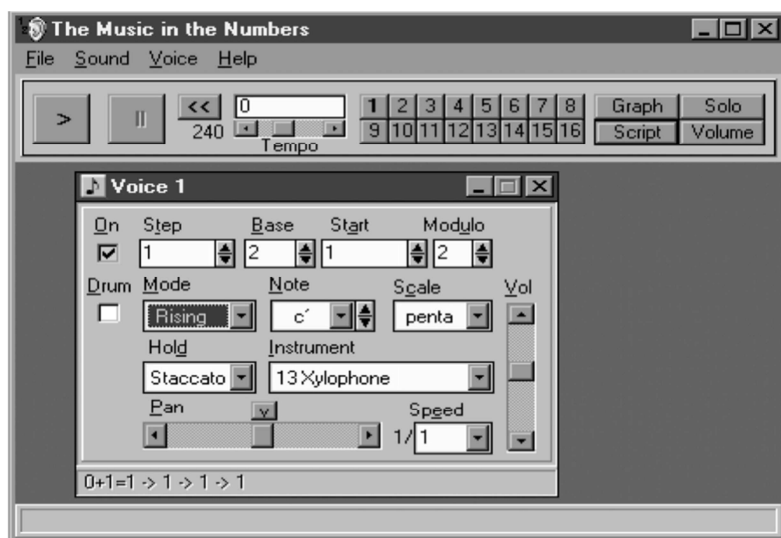
Програмата генерира редица от цели числа, на тяхна основа се формират фрактали, които се преобразуват в музика. Чрез диалогов прозорец, отварящ се със Script, се задава сценарий, който позволява да се изменят параметрите за синтез на музиката в процеса на изпълняване на композицията, което я прави по-динамична. Генерирането на гласовете се активира чрез On. Ако е установен Drum Mode, то гласът ще съответства на партията на един ударен инструмент. Мелодичните и ударните MIDI-инструменти се избират в разтварящия се списък Instrument. От списъка Hold се избира характерът на звучене: Normal, Pedal, Staccato. Регулаторът Pan установява вида на гласа, а Volume – силата му. В списъка Speed в частите на такта се избира продължителността на създаваната нота. Разтварящият се списък Note е предназначен за избор на началния звук: звук, който се изпълнява, ако в резултат на пресмятанията се получава 1. В списъка Scale се избира ладът, в който се създава композицията. Списъкът Mode позволява да се избира режим за създаване на гласа: All – нотата се създава за всяко отчитане на такта; New – ако след пресмятанията две съседни ноти са еднакви, те ще бъдат обединени в нота с удвоена продължителност; Same - нотата прозвучава само когато съвпада с предходната изсвирена нота; Rising и Falling - нота прозвучава само когато нейният тон е по-висок (по-нисък) от тона на предходната изсвирена нота. Генерирането на всяка партия става по една и съща постоянна формула, но параметрите на тази формула са достъпни за редактиране: Step – големина на стъпката, Base – основа на системата за изчисление, Start – задействане на началната стойност относно стойността на параметъра Note, Module – основа на системата на разликите.

Decimal number	Binary counter	Amount of 1s	Musical note
1	1	1	C
2	10	1	C
3	11	2	D
4	100	1	C
5	101	2	D
6	110	2	D
7	111	3	E
8	1000	1	C
9	1001	2	D
10	1010	2	D
...	etc



Фигура 120. Последование от тонове, съответстващо на числата от двоичния брояч [150]

MusiNum е чудесно средство за представяне на музикалния потенциал на себеподобността в числови редици (ако просвирилите всеки втори тон от последованието в горната таблица, резултатът ще бъде абсолютно същата мелодия). Можете да чуете два музикални примера, които съм генерирала с тази програма в *Приложение 4, папка MusiNum: MUSINUM.MID; MUSINUM2.MID*.



Фигура 121. Основен прозорец за композиция [150]

MUSICAL GENERATOR (ИТЕРАТИВНИ АЛГОРИТМИ)

Автор: Арнолд Райндър (Arnold Reinder); Musoft Builders

Shareware

Платформа: Windows

Поддържане на MIDI: Да

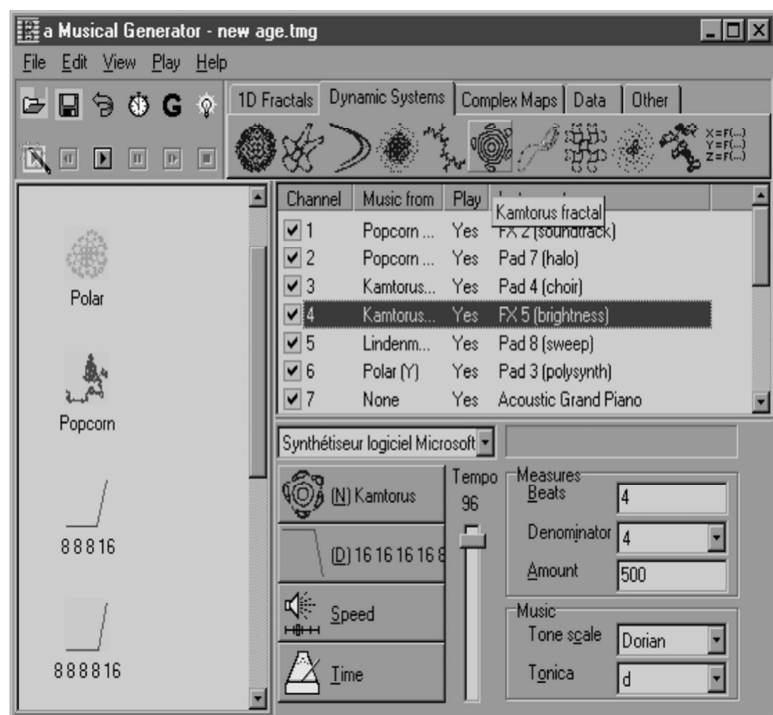
Наличност: <http://www.musoft-builders.com>

Програмата Musical Generator на Арнолд Райндър (*Arnold Reinder*) е програма, която интерпретира музиката от итеративни алгоритми, като фракталите и хаоса. Също така може да интерпретира музиката от графични образи, текст и числа.

Музикалният материал се получава от голям брой източници, включително хаотични карти (*Hopalong, Lorenz, Martin, Gingerbread Man, Henon, Polar attractor*); фрактали (*Mandelbrot set, Julia, Barnsley, Lambda, Newton*); шум (*white, Brownian, fractional brownian*); *Lindenmayer L-systems*; математически константи. Алгоритмите могат да се присъединяват независимо към всеки MIDI глас. Само на едно събитие се позволява да бъде композирано (един набор от алгоритми се преобразува към музикалните параметри). Програмата се състои от две основни секции (Фиг.122): секцията с данните и MIDI секция. Първата съдържа данните, от които музикалният генератор ще възпроизведе музикален материал (например фрактал), докато втората съдържа инструкции как да просвири музиката. В тази програма композиторът може да дефинира музикални тоналности, да контролира динамиката, дължината на пиесата и т.н. В основния прозорец програмата има няколко бутона, всеки от които съдържа групирани функции: *1D Fractals, Dynamical Systems, Complex Maps, Data*. Тези функции предоставят съдържанието на секцията с данни. Едно кликуване върху иконите ще представи функциите в секцията с данни. Веднъж след като функцията е избрана потребителят може да я редактира по различни начини. На разположение е и изрисувач инструмент, който да визуализира изходния резултат. За да се конвертират данните в музика, композиторът трябва да определи функциите от секцията с данни, които от своя страна контролират аспектите от MIDI секцията. Има четири музикални аспекта, които могат да бъдат контролирани от една функция: Тонове, Продължителност, MIDI сила (*speed, loudness*) и Време. Тези аспекти кореспондират на четири бутона върху MIDI секцията. Функциите се определят чрез провлачване на иконата им от секцията с данни върху един от тези четири бутона. Музикалният изходен резултат може да бъде представен чрез вграден piano roll и след това съхранен в MIDI файл за по-нататъшна

редакция в други приложения. Програмата също така има капацитета да се синхронизира с други пакети, чрез MIDI времеви часовник.

Можете да чуete няколко музикални примера, които съм създава с тази програма в *Приложение 4*, *nanka Musical Generator: Brown noise.mid; Fibonacci.mid; Kamtorus.mid; Kamtorus_Ginger_Rossler.mid; Lindenmayer.mid; Martin.mid; New Age.mid; Rossler.mid; Uniform noise.mid; White noise.mid*.



Фигура 122. Основен прозорец на Musical Generator [151]

CAMUS (CELLULAR AUTOMATA)

Автори: E.Miranda, Kenny McAlpine, Stuart Hoggar

Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://www.maths.gla.ac.uk/~km/dsysmus.htm>

CAMUS е система за композиране, използваща клетъчна автоматизация, развита и имплементирана за Windows от създателите си Едуардо Миранда (*E.Miranda*), Кени Макълпин (*Kenny McAlpine*) и Стюарт Хогар (*Stuart Hoggar*). Концепцията която лежи зад програмата е интерпретирането на музика от клетъчна автоматизация, чрез кодово-графично преобразуване на живи клетки от *Game of Life* към три тона. За да се установи кой инструмент ще просвири дадена клетка, системата наблюдава състоянието на

съответстващата клетка в *Demon Cyclic Space*. Инструментите, които просвирват музиката за всеки канал се определят чрез диалогова кутия на инструментите.

VOX POPULI (GENETIC ALGORITHMS)

Vox Populi е интерактивна композиционна програма, която използва генетични алгоритми, предложена от Йонатас Мандзоли (*Jonatas Manzolli*) и Артемис Морони (*Artemis Moroni*). В тази програма генетичните алгоритми се използват, за да развият поредица от акорди, популация от акорди. Всеки акорд от популацията има четири тона, всеки от които е представен от 7-битов низ. Следователно един акорд се представя от низ от 28 бита (1001011 0010011 0010110 00101010). Генетичните операции на пресичащи се мутации се прилагат към този код, за да генерират нова популация. Критерия за издръжливост има три фактора: мелодична, хармонична и гласова издръжливост. Мелодичната издръжливост се оценява чрез сравнение на тоновете от акорда с потребителски дефинирани относителни стойности. Тази относителна стойност определя вида на тоналния център или атрактор: най-близките тонове до центъра са близки до тази стойност, докато най-отдалечените са близки до стойността на издръжливост.

Хармоничната издръжливост взема под внимание консонанса на акорда, а гласовата издръжливост определя дали тоновете от акорда са в потребителски дефинирания диапазон. Потребителският интерфейс е много лесен за употреба и предоставя плъзгачи и други контролери за спецификацията на параметрите и музикалните атрибути. Програмата може да се използва и за изпълнение в реално време.

Графичният контрол е интересен от гледна точка на изпълнение, понеже предлага метафорични жестове на диригент, сякаш изпълнителя е диригент на оркестър. Мандзоли композира музика използвайки ръкавица, снабдена със сензори, които транслират движенията на ръката му върху този графичен пад.

HARMONY SEEKER

Harmony Seeker комбинира генеративните способности на клетъчната автоматизация със селективните механизми на генетичните алгоритми в среда за музикално композиране. Програмата е предложена от Валерио Таларико (*Valerio Talarico*), Евеонора Билота (*Eleonora Bilotta*) и Пиетро Пантано (*Pietro Pantano*). Основно програмата интерпретира изходния резултат на популация от едномерна

клетъчна автоматизация в MIDI данни, но докато не се постигне удовлетворителна нова генерация, системата предлага генетична технология, за да подобри изпълнението на популацията.

Философията зад системата е следната: една клетъчна автоматизация се счита за индивидуалност, чиито характеристики се определят от преходни правила и от първоначалното състояние на клетките; една и съща редица от правила ще произведе различен резултат, след като се приложи към отделни първоначални конфигурации на клетките. Системата стартира с популация на клетъчна автоматизация, всяка от които предоставя материал за изходния резултат и обективността им се изразява в това да подобрят изпълнението на тази популация, откривайки оптималните правила и конфигурации за клетките. Това се постига чрез средствата на генетичните алгоритми, като мутация и застъпване, където селектирания критерий се базира на постоянна функция.

Основният прозорец на програмата има различни бутони за настройка на параметрите на системата и наблюдение над изпълнението. Основната част е *The simulation tuning centre*, където потребителят избира естеството на постоянното измерване, метода за интерпретиране на MIDI данните и типовете генетични алгоритми; всеки един от тези аспекти има свои собствени дефинирани от потребителя настройки. Останалите бутони предоставят графика за изобразяване на еволюцията на популацията и средства за интервенция по време на еволюционния процес.

BRAIN INTERFACE: IBVA SYSTEM

Interactive Brainwaves Visual Analyser (IBVA) е система, която предоставя анализатор на човешкия мозък (Фиг.123). Програмата взима сигнали до 60 Hz от електродни сензори, поставени на главата на човека и изпълнява множество операции. Информацията от електродите може да бъде видяна на прозорец с Raw данни, подобно на конвенционалните електроенцефалограмни екрани, но IBVA също така изпълнява *Fast Fourier Transform* (FFT) анализ на този сигнал, за да го представи като редица от честоти и амплитудни стойности.

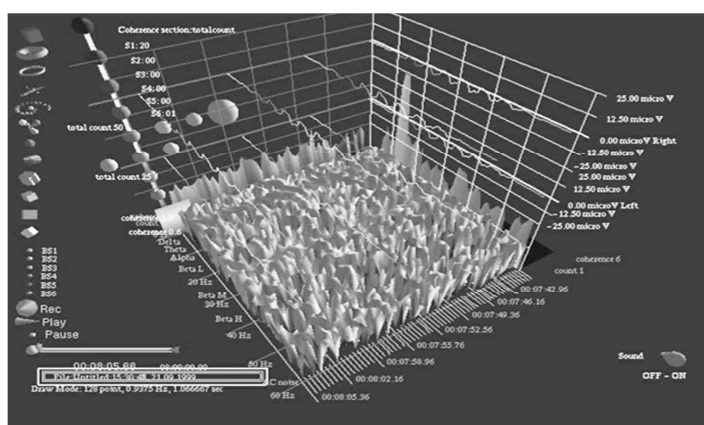
Интересна черта на програмата е че потребителят може да определи действия за различни честотни обхвати на анализирания резултат. Когато специфични честоти достигнат определена гранична стойност, тогава системата ускорява действието, свързано с обхвата, където честотите пропадат. Системата прихваща вълни от мозъчната дейност (*long-term coherent waves*), като *alpha*, *beta*, *delta* *theta* честоти (

обхващащи от 2 Hz до 45 Hz); също така прихваща ниско-честотните сигнали (под 2 Hz), свързани с движенията на окото.

Поради това е възможно да се асоциира движението на очите плюс честотните сигнали от мозъчната дейност, със специфични задачи, като например да контролира MIDI синтезатор. Системата има вградени ALPS+Orchestration модул, който улеснява тази спецификация. ALPS+ се намира около кодово-графичното преобразуване на MIDI тонове, което позволява контрола върху високо ниво музикални структури.

IVBA се предоставя с електроди за поставяне върху човешката глава и има малък трансмитер, който се прикачва към главата и след това свързва към компютър. Това също е много интересно от музикална гледна точка в случай, че композиторът би

пожелал да използва системата за живо изпълнение.



Фигура 123. Interactive Brainwaves Visual Analyser [152]

MIDI възможностите на програмата предлагат неограничени способности за композитора, особено когато се използва и друга система за живо изпълнение. Например, проф.Силвия Пенгили (*Sylvia Pengilly*) използва IVBA заедно със софтуера Мах, за да композира *Interface*. В тази творба, тоновете прихванати от сигналите на мозъчните вълни се изпращат към подпрограмата Мах, която разпределя навлизания тон в серия от нарастващи тонове в интервали от чисти квартали. Това създава основа за мелодичния контур при променливи спадове или възвишения на встъпващите тонове. Мах програмата е създадена така, че когато дейността на потребителския мозък достигне специфично ниво, се започва композирано последование от перкусионни звуци. Краят на това последование ускорява часовник, което позволява на основния образец да продължи за определено време. Разбира се звученето на пиесата зависи изцяло от активността на изпълнителския мозък. Идеята е изпълнителят да започне в притихнало състояние на ума, което предоставя ниски честоти и след това апостепенно да засили дейността на съзнанието си, което се изразява в по-високи честоти, докато не

започне перкусионната поредица. За да постигне този контрол, изпълнителят танцува или прави леки движения, което афектира върху нарастването на честотите.

ALGORITHMIC BUT LIVE - M

М е една от първите комерсиални програми за алгоритмично композиране, създадена през 80-те години от Дейвид Дзикарели (*David Zicarelli*), Джо Чабад (*Joe Chabade*), Джон Офенбах (*John Offenbach*) и Антъни Уидоф (*Antony Widoff*). Конкретно става известна на потребителите на Amiga и Atari, която е с най-добрата платформа за MIDI. Програмата позволява на компютъра да стане истински партньор по време на композиционния процес. Основно, потребителят вкарва музикални идеи и материал, които да се използват в пиесата, като мелодични фрази, вероятности за тонове, ритмични образци, динамичен обхват, темпо и тогава М ви оставя да контролирате как тези процеси ще се случат в реално време; има Conducting Grid, който позволява на потребителя да афектира върху операциите на цялата програма, по време на генеративния процес. М предоставя много механизми, които позволяват на потребителя да си взаимодейства с компютъра, така че различието между потребител и компютър почти да изчезне. Потребителят може да присъединява Midi тонове към специфични функции, които контролират програмата, правейки М отличен софтуер за употреба, във взаимодействие с алтернативни Midi контролери и интерфейси като системата IBVA.

GINGERBREAD MAN

Автор: Фил Томпсън (Phil Thompson)

Freeware

Платформа: Windows

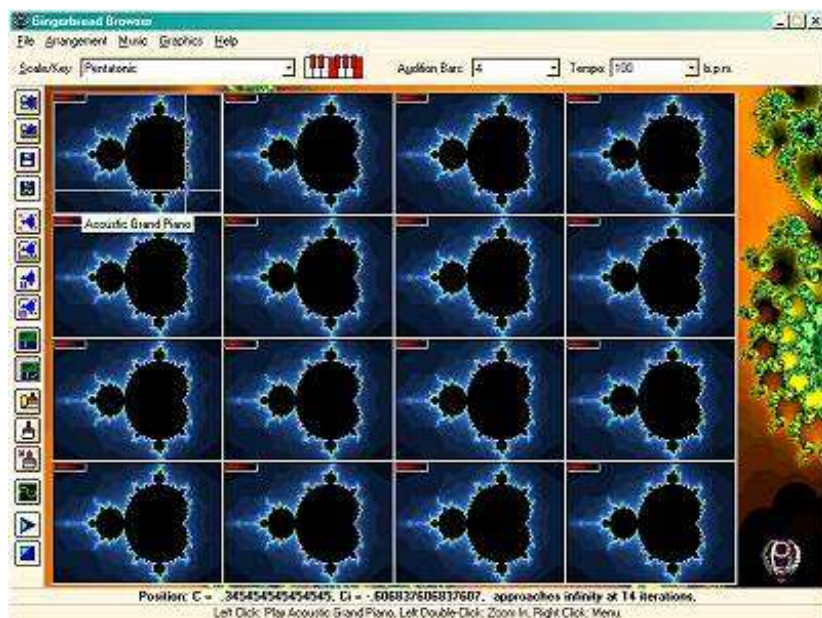
Поддържане на MIDI: Да

Наличност: <http://www.organised-chaos.com/oc/ginger.html>

Създателят на тази програма, Фил Томпсън, е един от най-популярните съвременни композитори на фрактална музика. (*Приложение 3, Phil Thompson - Strange Attractors; Phil Thompson-Shadows Fall; Steeltown*). Програмата, която той създава преобразува фракталните функции на *Mandelbrot* и *Julia* редиците към музикални параметри. Всеки MIDI глас има свой собствен прозорец, където потребителят може да манипулира и уголемява или намалява фракталния образ. След това потребителят може да избира определена секция от образа, от която да се генерира

след това музиката. Програмата позволява на композитора пълен контрол над музикалните параметри, като тоналност, динамика, трайност, темпо и т.н.

Основната част от работата в програмата се извършва в прозореца Gingerbread Browser (Фиг.124).



Фигура 124. Gingerbread Browser [153]

Екранът се състои от 16 Манделброт фрактали. Всеки един фрактал се преобразува към свой собствен инструмент. В началото всеки инструмент е с тембър на акустично пиано, с изключение на 10 –ти канал, който е запазен за ударната секция. За да се открие мелодия, може да се кликне върху разположения най-горе и вдясно фрактал. Кликвайки върху отделни точки от този фрактал, ще възпроизведе различни мелодии. Самата графика е като карта, като всяка точка от нея генерира различна мелодия. Точки, които са близки една до друга, създават сходни мелодии. Съответно точки, които са отдалечени една от друга, създават различаващи се мелодии. Най-интересните мелодии се създават в дясната част на фрактала, а най-минималните, ритмични образци – в лявата част. Най-дългите фрази се откриват на граничните области на фрактала (Фиг.125).



Фигура 125. Гранични области от фрактала [153]

Ако откриете мелодия, която ви харесва, но искате да я доразвиете, можете да увеличите мащаба на фрактала, в точката която желаете, за да постигнете мелодията, която търсите (Фиг.126).



Фигура 126. Увеличение мащаба на фрактала [153]

Когато откриете желаната мелодия, просто кликвате с десен бутон върху съответната точка и избирате Store Coordinate. Веднъж след като едта координата е съхранена, в левия горен ъгъл на фрактала ще се появи малко зелено прозорче. То индикира, че вече има съхранена координата.

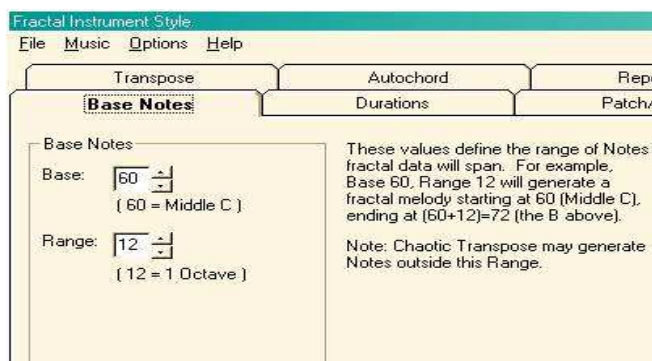
След като намерите мелодия, която да звучи приятно с тембъра на акустично пиано, може да продължим с търсенето на басова линия. Това може да се направи чрез дясно кликване върху първия фрактал и избиране на Copy Acoustic Grand Piano към » 2 Acoustic Grand Piano (Фиг.127). Падащ прозорец ще изиска от вас да изберете кои аспекти от първия фрактал да бъдат копирани към втория.



Фигура 127. Избор на копиращ инструмент [153]

В този случай първия фрактал се копира върху втория. Това означава, че вторият фрактал ще има същата мелодия като първия, но целта на тази операция е към втория фрактал да се добави басова линия. Кликвайки с десен бутон върху втория фрактал може да изберете Edit Fractal Instrument Style.

Ще се появи прозореца Fractal Instrument Style (Фиг.128), който съдържа 8 различни секции, в които могат да се променят мелодиите и инструментите.

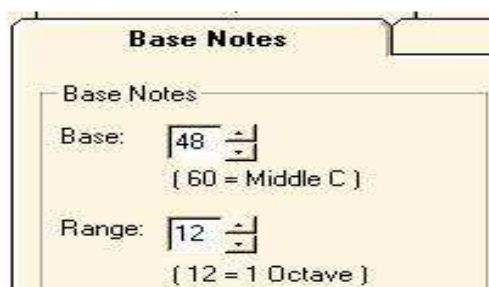


Фигура 128. Fractal Instrument Style [153]

Първо ще използваме Base Notes бутона, за да направим басовата линия да звучи една октава по-ниско от тази на пианото.

По принцип мелодията ще бъде генерирана от тонове, които взимат за основен тон (Base Note 60) или до от първа октава (Middle C), с обхват от 12 тона (октава) над него. Така че всеки тон в мелодията ще бъде между тона 60 и тон $(\text{Base } 60 + \text{Range } 12) = 72$.

За да се направи басовата линия една октава по-ниско, вадим 12 тона от Base Note, т.е. $60 - 12 = 48$. Така променяме стойността от 60 на 48 (Фиг.129).



Фигура 129. Линията на баса [153]

А сега нека направим басовата линия по-бавна от тази на пианото. Изберете бутона Durations и променете Variance drop-down кутията от Busy Melodic на Slower Melodic. Сега нека покажем на Gingerbread, че това е бас инструмент, а не пиано. Изберете Patch / EFX бутона и използвайте падащото меню, за да изберете акустичен бас. Във всеки един етап можете да прослушате промените, които правите, избирайки Play Instrument от менюто Music.

След като вече е създадена басовата линия, можете да потвърдите промяната чрез File » OK от Music менюто.

За да чуете едновременно пианото и баса, кликнете с десен бутон върху който и да е фрактал и изберете Play All Stored Instruments или кликнете върху бутона Play Composition.

По принцип когато просвирвате композицията, тя ще прозвучи само в 4 такта. Това може да се промени променяйки стойността в Audition Bars. Можете да промените тази стойност до 128 такта, което ще възпроизведе много по-дълга композиция. 128 такта се максималната дължина на пиесата, която може да генерира Gingerbread. Можете също така да забързате или забавите темпото на цялата композиция с настройките за Tempo.

Можете също така да промените Лада/Тоналността (Scale/Key) на пиесата (Фиг.130).



Фигура 130. Промяна на Лад/Тоналност [153]

Ако искате да съхраните композицията си, изберете File » Save Project As, което ще съхрани един GIN file, което типа проект на програмата Gingerbread. Можете също така да експортирате композицията си като стандартен MIDI файл (.MID). Изберете Export Midi As — в този случай ще имате възможност да определите броя тактове, които да се експортират (максималния брой е 128). След като експортирате в midi формат, можете да намерите композицията си в c:\program files\gingerbread\midi\ folder. Пример за фрактална композиция, генерирана с тази програма можете да чуete в **Приложение 4, Gingerbread.mid**.

MANDELBROT MUSIC PROGRAM

Автор: Yo Kubota

Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://www.fin.ne.jp/~yokubota/>

Програмата се базира изцяло на редицата на Mandelbrot. Програмата преобразува итерационни стойности към потребителски дефинирани музикални параметри.

CHAOS VON ESCHENBACH

Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://www.cisnet.or.jp/home/magari>

Програмата използва редицата на Mandelbrot, Julia, white noise и логични уравнения. Притежава огромен набор от преобразуващи схеми.

LOSHUMUSIC И FIBONACCIBLUES

Платформа: Macintosh

Shareware

Тези две приложения за Мас използват редицата на Фибоначи и алеаторни процедури, като база за генериране на музика . Може да се изтеглят от

<ftp://mirror.apple.com/mirrors/Info-Mac.Archive/art/fibonacci-blues-02.hqx>

<ftp://mirror.apple.com/mirrors/Info-Mac.Archive/art/loshu-music-02.hqx> respectively.

SYMBOLIC COMPOSER

Автор: Tonality Systems

Commercial

Платформа: Macintosh

Наличност: <http://symcom.hypermart.net>.

Symbolic Composer е комерсиална програма, която използва собствен език от 700 вградени блока за композиране. Използва стохастични процеси, хаотични карти, фрактали, числова теория. Позволява да се прилага голям обхват от алгоритми . Много от алгоритмите, които не са включени в програмата, могат да бъдат имплементирани (програмирани), използвайки собствения език на програмата. Програмата не е лесна за употреба или изучаване, но в ръцете на способни композитори е един от най-мощните инструменти за експериментална музика. Можете да чуете три музикални примера, генерирани чрез тази програма в *Приложение 4, папка SCOM: Bridge.mid*;

INVENTION1.mid; INVENTION2.mid). Самите кодове, чрез които са генерирани тези музикални примери, можете да видите в *Приложение 5, Code for Invention1 and Invention2; Code for bridge*.

CSOUND

Freeware

<http://mitpress.mit.edu/e-books/csound/fpage/getCs/getcs.html>

Csound е отворен компютърен език, създаден за музикално композиране и звуков дизайн. Толкова е разностранен, че позволява имплементацията на който и да е алгоритмичен процес. Езикът за програмиране Csound може да бъде добро средство за създаване на експериментални данни към звукови параметри. Изобретателят на Csound - Ханс Микелсън (Hans Mikelson) , публикува няколко изумителни Csound файлове , които използват различни елементи от фракталната графика, като параметрични стойности за Csound инструменти. За пример, един от тези файлове, използва броя на повторенията или loops в един образ, за да определи спектъра на неговия звук, а друг файл изчертава RGB стойностите на един цветен образ към честотния им еквивалент . [www.werewolf.net/~hljmm/Ezine/synthesis].

FMUSIC

Автор: David Singer

Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://members.aol.com/dsinger594/caman>

Тази програма преобразува Lindenmayer L-System и едномерна cellular automata към музикални параметри, дефинирани от потребителя. Генерира петгласова полифония. Потребителят има пълен контрол как се изпълнява кодово-графичното преобразуване.

QUASI FRACTAL COMPOSER

Автор: Пол Уейли (Paul Whally)

Shareware/Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://www.randomtunes.com>

Програмата използва стохастични и детерминистични методи. Параметрите за тези процеси се определят от потребителя. Инструментариум, тоналности, трайност на нотите, динамика и други музикални параметри също могат да се дефинират от потребителя.

Микеланджело веднъж забелязва, че всеки мраморен блок съдържа в себе си скулптура, която очаква да бъде открита от даден творец. Тази програма възплъщава идеята, при която се генерират блокове от музикален мрамор или мозайка, която авторът да оформи и да ѝ придаде образ.

Квазифракталността е плаващо понятие, с което се показва самоподобността в различни дискретни скали.

Квазифракталната музика всъщност е съчетание на фрактална и сюрреалистична музика, като е крайно циклична. Използва алгоритми, за да спазва принципите на традиционната музикална структура.

Melody-to-Structure – тази функция позволява да се превеждат лесни мелодии от 3 до 16 ноти в Structure. В самия прозорец се въвеждат слоговите наименования на нотите – напр. A, A#, B, C, C#, D, D#, E, F, F#, G, G#. след което преводът ще се покаже в прозореца на Structure .

Structure е низ от 3 до 16 цифри. Този низ е базата на самоподобността на музиката, като директно или индиректно контролира повечето от параметрите в композицията. Единственото условие е че цифрите не трябва да са еднакви (т.е. "333333" , докато "333334" е правилен избор).

Mapping е цяла стойност от 1 до 32767. Контролира произволният числов генератор на програмата . Ако Mapped Parameters Auto е избран, на всички Mapped Parameters ще бъдат присъединени произволни стойности.

Една QFC композиция съдържа "n" на брой тракове, като "n" е броя на тракете, който е избран в Tracks контрола. Всеки трак се разбива на "x" периоди , където "x" е броя на цифрите в Structure. Всеки период ще съдържа от 1 до "x" на брой фрази.

Structure е сърцето на композиционния процес. Рефлектира навсякъде от цялостната комп.форма до настройката и трайността на индивидуалните ноти. Structure управлява композиционния процес чрез Mapped Parameters. Всеки Mapped Parameter има присъединена една от 28 възможни структурни карти. Структурната карта

определя как Mapped Parameter да преведе всяка една цифра от Structure в контролни стойности за специфичен композиционен процес или характеристика.

По време на композиционния процес, Mapped Parameters със структурна карта от 1 до 4 , се изчислява от началото на всеки период и остава в действие през целия период. Mapped Parameters със структурна карта от 5 до 12 се изчислява от началото на всяка фраза . Mapped Parameters със структурна карта от от 13 до 28 се изчислява за всяка индивидуална нота. Така че – карти от 1 - 4 могат да изпълняват контрол над формата на композицията , карти от 5 - 12 могат да контролират на по-детайлно ниво и карти от 13 - 28 контролират определението на всяка една нота.

Ако в прозореца на Structure се въведе последователност, който повтаря началото - 0480022440480, това ще има "въртящ" ефект в композицията. Друг вариант е да се повтарят цифри - напр. 00, 22 и 44 . Това придава ритмичен ефект на мелодията, но се получава пренасищане от повторения.

Можете да чуете няколко музикални примера, които съм създала чрез тази програма в *Приложение 4, нанка Quasi Fractal Composer: first.MID; full1.MID; orch1.MID; SADBA.MID; second.MID.*

LORENZ

Автор: Джоуз Наваро (Jose Navarro)

Shareware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://globalia.net/janc>

Тази програма прилага функции на Lorenz attractor, за да генерира последования.

MAKE PRIME MUSIC

Автор: Араманд Тюрпел (Armand Turpel)

Freeware

Платформа: MS-DOS

Поддържане на MIDI: Да

Наличност: <http://www2.vo.lu/homepages/armand/index.html>

Редица на простите числа. Прилагат се три различни преобразуващи схеми върху редиците.

GENETIC SPECTRUM MODELING PROGRAM

Автор: Рей Джъргънс (Ray Jurgens)

Freeware

Платформа: MS-DOS

Поддържане на MIDI: Да

Наличност: <http://autoinfo.smartlink.net/ray>

Тази програма комбинира cellular automata, genetic algorithms и 1/f noise, за да генерира фрактални мелодии. Потребителят има контрол над тоналностите, които се използват, броя на клетките (за процесите на cellular automata), броя на генерациите, различните вероятности настройки, като мутационен размер и т.н. Програмата генерира на произволен принцип поредици от тонове, които са напълно несвързани. Поредиците от тонове представят група от клетки, които следват правилата на Cellular Automata. Всяка генерация генетично се модифицира, за да превърне клетките в специфична спектрална форма, която имитира 1/f noise. Този процес включва случайна мутация на клетките. Програмата генерира само един глас за поредица от параметри.

THE WELL TEMPERED FRACTAL

Автор: Робърт Грийнхаус (Robert Greenhouse)

Shareware

Платформа: MS-DOS

Поддържане на MIDI: Да

Наличност: <http://www-ks.rus.uni-stuttgart.de/people/schulz/fmusic/wtf>

Програмата използва Chaotic attractor (Hopalong, КАМ torus, Mira, Chebychev function, Julia map). Потребителят контролира музикални параметри, като тоналност, диапазон на тоновете, трайност и т.н.

LMUSE

Автор: Дейвид Шарп (David Sharp)

Freeware

Платформа: MS-DOS, Java

Поддържане на MIDI: Да

Наличност: <http://www.geocities.com/Athens/Academy/8764/lmuse/lmuse.html>

Програмата използва Lindenmayer L-System. Генерира графично представяне на l-системно правило, от което се получава и музиката. Потребителят може да дефинира

много различни типове на кодово-графично преобразуване. Генерира се само една композиция за една правило. Потребителят може само да реши тоналността, от която да се получават тоновете.

SOFTSTEP

Автор: Algorithmic Arts

Commercial

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://www.algoart.com/software/index.htm>

Програмата използва фрактали от поредиците на Mandelbrot и Julia; хаотични атрактори (Mira, Martin, Henon, Hoptalong); стохастични процеси, числова теория.

RANDOM PHASE MUSIC GENERATOR

Автор: Tak-Shing Chan

Freeware

Платформа: Linux

Поддържане на MIDI: Да

Наличност: <http://www.engr.newpaltz.edu/~chan12/phase.html>

Музиката се генерира чрез фазиране на случайно генерирани образци - процесът представлява повтаряне на едни и същи образци, с малки разлики в скоростта им, така че те да излизат едва забележимо от синхронизация. Потребителят има контрол над музикални параметри като тоналности, продължителност, темпо, както и над дължината на повтарящия се образец.

SILENCE

Автор: Майкъл Гогинс (Michael Gogins)

Freeware

Платформа: всички платформи

Поддържане на MIDI: Да

Наличност: <http://www.pipeline.com/~gogins/>

Програмата използва Lindenmayer L-System, итерационни функционални системи, хаотични атрактори. Тази програма изисква инсталацията на програмата CSound.

AC TOOLBOX

Автор: Пол Бърг (Paul Berg)

Freeware

Платформа: Windows

Поддържане на MIDI: Да

Наличност: <http://www.koncon.nl/ACToolbox>

Програмата предоставя възможности за създаване на събитийно музикално пространство чрез стохастични функции, хаотични системи, рекурсии. Генерира CSound и Opcode MAX файлове.

COMMON MUSIC

Автор: Рик Таубе (Rick Taube)

Freeware

Платформа: Windows, Macintosh, MS-DOS

Поддържане на MIDI: Да

Наличност: <http://www-ccrma.stanford.edu/CCRMA/Software/cm/cm.html>

Common Music е обектно-ориентирана среда за музикално композиране. В програмата могат да се имплементират много алгоритмични процедури: стохастични процеси, хаотични системи, числова теория.

KEYIT

Freeware

Платформа: Windows, Linux

Поддържане на MIDI: Да

Наличност: <http://thompsonresidence.com/keykit>

KeyIt е програмен език и графичен потребителски интерфейс за MIDI. Потребителят може да имплементира много алгоритмични процеси, включително стохастични процеси, фрактали, хаотични карти.

REAL TIME COMPOSITION LIBRARY

Автор: Карлхайнц Есъл (Karlheinz Essl)

Freeware

Платформа: Macintosh

Поддържане на MIDI: Да

Наличност: <http://www.essl.at/works/rtc.html>

RTC-Lib е колекция от плъгини за системата Opcode MAX. Имплементира алгоритмични процедури, като стохастичните процеси, числови функции.

THE HIERARCHICAL MUSIC SPECIFICATION LANGUAGE

Автори: Фил Бърк, Лари Полански, Дейвид Розенбуум (Phil Burk, Larry Polansky, David Rosenboom)

Commercial

Платформа: Macintosh

Поддържане на MIDI: Да

Наличност: <http://www.softsynth.com/hmsl>

HMSL е програмен език за експериментално музикално композиране. Позволява на потребителя да имплементира алгоритмични процеси, като стохастичните, фракталните и числовите.

MAX И OMAX

Автор: Милър Пукете (Miller Puckette)

Commercial, Opcode Systems

Платформа: Macintosh

Поддържане на MIDI: Да

Наличност: <http://cycling74.com/products/maxmspjitter/>

MAX е програмна среда за музика. Нейната обектно-ориентирана програмна среда може да създава безкрайни варианти на персонални приложения. Който и да е алгоритмичен процес, който можете да си представите, е в състояние да бъде имплементиран в MAX.

Машинната импровизация обикновено се отнася за представянето на времево-базирани медиа данни, като музиката или се отнася за стилистично обучение, с ясно формулирани правила и прилагане на машинно заучени методи. Стилистичното обучение на музикални стилове използва статистически модели на мелодии или полифонии, за да създаде варианти на музикални примери. Последни опити в обучителните техники индикират, че могат да се емулират особени поведения и тези правдоподобни образци могат да бъдат възпроизведени от компютър за специфични области.

В областта на музикалната импровизация с компютри има забележими опити в статистическото музикално моделиране, което позволява прихващане на стилистични музикални правила по начин, позволяващ музикално съдържателно взаимодействие между човека и компютъра. Сътрудниците към OMax са експериментирали с няколко от тези модели през последните години и неотдавна показват резултатите от техните изследвания в OMax, една среда, която има предимството, че ползва мощта на OpenMusic, за моделиране и програмиране, и на MaxMSP, за представяне и работа в реално време.

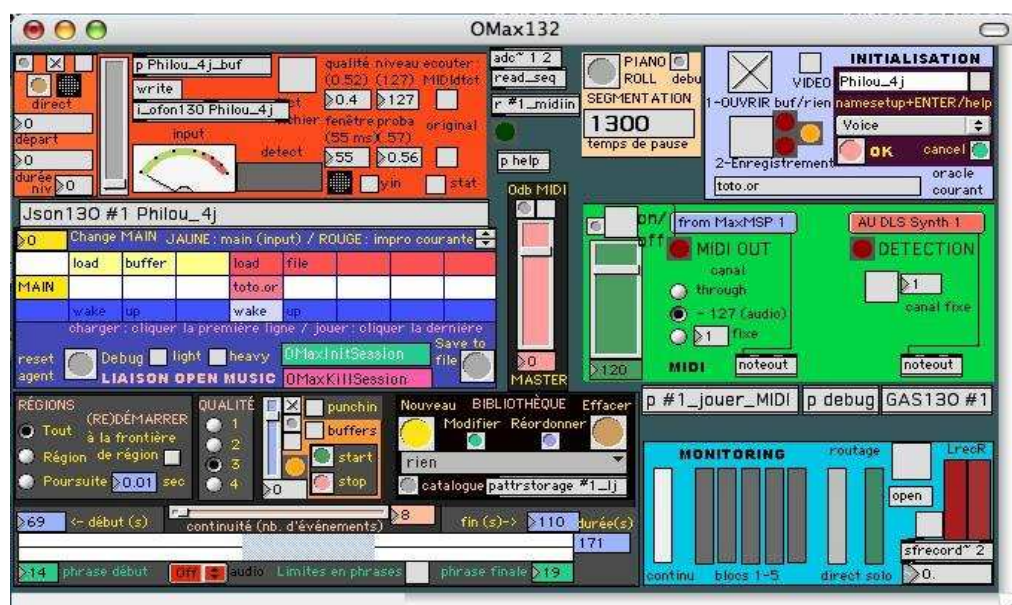
OMax позволява в реално време взаимодействие с човек, който свири, като генерира виртуална импровизация, стилистично обучение, стилистично моделиране. Работи на две отделни нива: едно е Max, което е близко до реалното време и включва бърза реакция, а второто ниво е OpenMusic, което има дълбок анализаторски и предсказващ поглед над миналото и бъдещето. Тези две концепции на времево взаимодействие и синхрон между комуникационни канали (OSC), са тези през които музикалните данни циркулират като сигнали и в двете посоки. В областта на OM, е лесно, дори докато системата работи да се смени тялото на една lisp функция и да се тестват вътрешните промени. Лесно е да се обогати генерацията чрез свързване на системата с различни композиционни алгоритми, които са на разположение в тази среда. Същото важи и за областта на Max, с неговата всеобхватна колекция от генерация в реално време и обработващи модули. Можем да възприемаме тази среда като независим модулатор и разширена експериментална среда за тестване на нови идеи.

OMax предоставя един виртуален музикален партньор, който взема всичките си познания от изпълнението на музиканта, което е съсредоточено във възпроизвеждането в реално време на аудио и Midi. Един начин да се разбере връзката, която свързва музиканта и OMax е да се характеризира като процес на стилистично инжектиране. Музикалната хипотеза, която стои зад стилистичното инжектиране е че един импровизиращ изпълнител е образован последователно от няколко източника. Изпълнителят слуша своите партньори. Той също така се вслушва в своето собствено изпълнение и мигновенното решение, което той създава между първоначалните намерения, може да се промени и да отвори нова посока. Звуковите образи на настоящото му изпълнение и тези, които са на другите изпълнители се съхраняват, така че да се носят в паметта между настоящето и миналото. Могат да се използват образци от паметта, като допълнителни източници на вдъхновение, което евентуално да бъде

комбинирано, за да създаде нови образци на импровизация. Идеята която стои зад стилистичното инжектиране е да действа, използвайки компютъра като външна памет, като това създава музикални фигури от миналото в една прекомбинирана форма, предоставяйки подобна, но винаги иновативна реконструкция на миналото. Интересно е да се наблюдава как човешкия партньор реагира на своя клонинг и променя своята импровизация спрямо тази на своя партньор. OMax слуша изпълнението на своя музикален партньор чрез Midi или аудио канали. По този начин извлича черти от високо ниво от постъпващия сигнал и го разделя в събития и фрази. OMax поддържа постоянно OMaxLisp (компонента OM/Lisp) с поток от Midi-подобна информация. От своя страна OMaxLisp непрекъснато изгражда памет за модела на последованието от събития. Този модел се нарича Oracle, базиран на работата върху редица от низове. OMaxLisp също така непрекъснато импровизира, което преглежда модела, за да се генерират различни поредици, което изпраща непрекъснат Midi поток към OMax.

OMax е софтуерна среда, която може да учи в реално време типични черти на музикален стил и да свири интерактивно заедно с музиканта, давайки аромата на машинната импровизация. Базира се на изследвания върху стилистични модели, както и на изследвания върху импровизацията с компютър.

OMax използва оригинални варианти на алгоритъма Factor Oracle, специално адаптиран за музикална обработка. OMax поддържа Midi, аудио и видео данни (Фиг.131).



Фигура 131. OMax позволява семантично ниво на представяне на сесията и рекомбинация и трансформация на този материал в реално време. [154]

В примера даден в *Приложение 4, PhilouOMaxShort.mp3*, OMax свири със саксофониста Филипе Леклер (Philippe Leclerc). Върху единия канал (първият, който ще прозвучи в този запис) е живия източник. На другия канал е изпълнението на OMax. OMax започва да свири толкова скоро, колкото му е нужно, за да научи няколко ноти от живото изпълнение на музиканта, затова започва с тремоло. Впоследствие моделът се разширява, обогатява и започва да клони към дуо.

В останалите музикални примери, дадени в Приложение 4, могат да се чуят още няколко примера за импровизация между инструменталисти и програмата OMax. (*Приложение 4, jaco.omaximpro.small.mp3; OMaxGarsonImpro1.mp3; OMaxGarsonImpro2.mp3; ricercar.mp3*)

CMIX И RTCMIX (REAL TIME CMIX)

Автори: Пол Лански, Ланс Граф, Дейв Мадол, Брад Гартън, Дъг Скот, Ерик Лион (Paul Lansky, Lance Graf, Dave Madole, Brad Garton, Doug Scott, Eric Lyon)

Freeware

Платформа: IRIX, Linux

Поддържане на MIDI: Да

Наличност: <http://music.columbia.edu/cmix>

Cmix е компютърен език, създаден за звуков синтез, както и за музикално композиране. RTCmix е версия на Cmix в реално време, където контрола върху инструментите може да бъде направен в реално време. Използвайки RTCmix езика може да бъде имплементирана която и да е алгоритмична процедура.

RUBATO

Автор: Крис Там (Chris Tham)

Freeware

Платформа: MS-DOS

Поддържане на MIDI: Да

Наличност: http://members.value.com.au/christie/rub_full.html

Rubato е програмна, нотираща и изпълнителска среда за музика. Позволява имплементацията на алгоритмични процедури.

LSCORE

Автор: Сенаст Андрад (Senast Andrad)

Freeware

Платформа: всички платформи

Поддържане на MIDI: Да

Наличност: <http://listen.to/algo-comp>

Lscore е имплементация на L-System за CSound. Изисква инсталацията на пакета CSound.

NYQUIST

Freeware

Платформа: Macintosh, Windows

Наличност: [http://www.cs.cmu.edu/afs/cs.cmu.edu/project/music/web/music.softw
are.html](http://www.cs.cmu.edu/afs/cs.cmu.edu/project/music/web/music.softw
are.html)

Nyquist, открит от Роджър Данънбърг (*Roger Dannenberg*) е програмен език за музикално композиране и звуков синтез – поддържа едновременно високо ниво на композиционни задачи, както и долно ниво на сигнална обработка в една интегрирана среда. Повечето звуково синтезиращи езици (най-популярния от които е CSound) не са изцяло подходящи за алгоритмично композиране, защото са ограничени откъм дизайна на инструментите – даден тон, който трябва да бъде просвирен от определен инструмент трябва да бъде указван ръчно тон по тон.

Nyquist е имплементиран в програмния език Lisp, по-специално в Xlisp, едно разширение на Lisp, което е напълно безплатно за некомерсиални цели. Nyquist може да бъде считан като библиотека на функциите на Lisp, които могат да бъдат извиквани чрез допълни функции в програмата. Xlisp бива разширен, за да внедри звуци, като вградени типове данни, което дава на Nyquist огромен потенциал. По този начин композиторите пишат техните алгоритмични програми в Nyquist, докато пишат Lisp програми. Предимството на тона е че композиторите имат пълния потенциал на Lisp, който е мощен език използван в изследванията на изкуствения интелект, комбиниран с широка вариантност от функции и абстрактни структури за композиране, създадени за Nyquist.

ПРИНОСИ

В тази глава от дисертацията съм представила и анализирала едни от най-известните софтуерни продукти до момента, които се използват за генериране на алгоритмична музика. Програмите и програмните езици, които са представени в тази глава от дисертацията не са всички компютърни софтуерни средства, които съществуват до момента. Всеки ден се създават нови и нови програми, други се актуализират и обновяват. Някои от URL адресите дадени към всяка една програма, може вече да не са актуални и в този случай ще е добре да потърсите съответната програма чрез някоя Internet търсачка.

Многообразието, което тези софтуерни продукти предлагат се ограничава до необходимостта използвания продукт да бъде със следните качества: бързо действащ, с добра вариантност, разнообразие от алгоритми и възможност за MIDI управление. Използвайки тези оптимални характеристики избрах няколко програмни продукта, измежду които са *FractMus*, *Musical Generator*, *MusiNum*, *Quasi Fractal Composer* и *SCOM*, чрез които създадох алгоритмични композиции, чиято музикална реализация съм предоставила в **Приложение 4** и описала към всяка една програма, чрез която съм генерирала съответната алгоритмична композиция.

ЗАКЛЮЧЕНИЕ

Музиката е вид информация, съставена от звуци, които могат да се представят като поток от данни. Езикът на музиката се изучава години наред и след като се изучи се разбира, че е опозната само една частица от същността му. Компютърните технологии са средството, чрез което музиката започна да се превежда и на други езици - започна да се програмира, визуализира, дигитализира. Чрез компютъра музиката освен като абстрактна величина придоби значението на универсален език, който интегрира и синтезира в себе си похватите на математиката, клон от която са уравненията за фрактална геометрия. Музиката започна да говори с компютри, синтезатори, секвенсери и започна да изглежда различно - под формата на MIDI събития и фрактални визуални модели.

Компютърът е един инструмент за реализирането на абстрактни конструкции, които впоследствие да бъдат приложени в музиката, съобразявайки се с креативното мислене и вдъхновение на композитора. Използван благоразумно и съобразно с естествените параметри на неговите умения, компютърът позволява на композитора да експериментира с музикалните възможности на различни алгоритмични процедури и да оценява тяхното потенциално развитие в квалифицирани композиционни системи. Компютърът в алгоритмичната музика е най-подходящо приложим като композиционен инструмент, отколкото само като едно музикално решение.

При алгоритмичната музика композиторът борава със заложили неизменни алгоритми. При традиционните похвати на композиране авторът предварително знае и задава лада на творбата си, докато когато създава фрактална музика е ограничен от самия алгоритъм. Но в този случай отново имаме избор, защото можем да изберем самия алгоритъм, който да отговаря на търсенията ни, както и можем да зададем основни параметри, които този алгоритъм да използва при генерирането на музика.

Много често хората забравят, че в музиката се съдържа голяма част абсорбирана доза от математиката. В теорията на Питагор, числата са били основна част от музикалното преживяване, като мистерията на хармонията е решена и проектирана благодарение на универсалните математически коефициенти.

Музиката е език, доста различен, но вероятно подобен като форма в сравнение с разговорния такъв.

Както стана вече известно до средата на 80-те години на 20 век се е отдавало повече внимание на тембрите, отколкото на структурата на композираната музика. Това е разбираемо, поради ограничените изчислителни възможности на тогавашните компютри и относителната липса на богати звукови източници. Поради ограничените възможности на тогавашните компютри изглежда съвсем основателно, че повечето от алгоритмите за композиране са били концентрирани около логическото свързване на всеки тон със следващия, което се осъществява на принципа на вероятностните вериги на Марков. Но този принцип на композиране води до пренебрегване на по-големите структури като фрази или разнообразен тематичен материал.

Друга интересна област на изследване, станала неизменна част от настоящия дисертационен труд е свързана с фракталите, които осигуряват изключително интересни музикални резултати и тази област на изследвания стана една от най-вълнуващите области от проучването на новата музика . Създателите на фрактална музика се окуражават да публикуват техните творби и да си взаимодействат в търсенията си с физиката, математиката, алгоритмите и изкуствения интелект.

Теорията за фракталната музика обсъжда възможностите за потенциална употреба на рекурсия, итерация и комплексна математика, като вид разширяване на традиционната музика и на композиционните практики .

Трябва да осъзнаем, че тези фракталните образи, които са толкова естетически приятни, са получени от проста процедура и всъщност са само редица от числа. Кодово-графичното преобразуване на редицата на Манделброт към музика се оказва, че е по-сложно отколкото кодово-графичното преобразуване на хаотични атрактори. Всяка итерация от уравненията формулира хаотична система, кореспондираща към точка в кореспондиращата система на хаотичния атрактор. Напълно достатъчно е да се изчертаят няколко хиляди итерации, за да се достигне формата на атрактора. Кодово-графичното преобразуване на хаотични атрактори към музика се състои от кодово-графично преобразуване на всяка итерация от уравненията към музикално събитийно пространство. От друга страна, във фракталния образ, за всяка точка от образа, неговото генериращо уравнение трябва да се повтори множество пъти (обикновено хиляда), за да определи дали точката се изплъзва от безкрайността или не. По тази причина, изчисленията във фракталите нарастват хилядократно.

Като източник на изключително богата палитра от звуци могат да се използват похватите на фракталната геометрия, в съчетание с параметрите на обектите от виртуалната сцена. Трябва да се има предвид възможността за два подхода за

генериране на фрактална музика. Единият подход е използване на алгоритми за директно генериране на музика, докато при втория вече съществуващи фрактали се преобразуват към аудио параметри.

Ако съпоставим един фрактал в чистата му геометрична форма с музиката и създаването на фрактална музика, всъщност ще достигнем до извода, че той представлява самата основа – формулата, под формата на един безкраен, повтарящ се алгоритъм, на базата на който се създава музика, което пък сравнено с класическите похвати за композиране – е идентично с процеса на композиране в традиционната музика. В музиката и в нейните основни композиционни техники винаги има заложен замисъл или т.нар. Лайт мотив, който без значение колко е сложен и какво съдържание носи в себе си – подлежи на множество разработки и надграждания, в които вглеждайки се или по-точно вслушвайки се – откриваме малкия, първоначален мотив, станал основа за създаването на цяла симфония.

Вследствие на направеното изследване в настоящата дисертация, стигнах до заключението, че L-системите са популярен избор за композиционни алгоритми, понеже те генерират себеподобни структури, т.е. сходствата нарастват между микроскопичните и макроскопичните аспекти на една възможна композиция. Музикалната интерпретация на L-системите се състои в присъединяване на музикално значение на символите в произведения низ. Употребата на L-системите е интересна за композиторите по две причини. Първата е че много често сложните нарастващи процеси могат да бъдат представени от конструкцията на много прости производни функции. Втората е, че се получават нарастващи, непредсказуеми, себеподобни форми, което не би могло да се постигне от обикновени композиционни методи.

Веригите на Марков осигуряват много по-рафиниран механизъм за контрол на вероятностите. Използването на веригите на Марков, позволява на композитора да контролира редиците от събития, правейки вероятността на дадено събитие да зависи от предишното. Връзката между събитията във веригите на Марков проявяват различни възможности, съобразно с това как техните вероятности са подредени в матрицата. Системата на Шилингер засяга основно ритъма като структурално средство, докато Марков засяга основно интерваловите съотношения.

Подходът на композитора Леджарън Хилър да формира се характеризира от традиционен начин на мислене. В неговите ранни компютърно-генерирани композиции, чийто брой е измежду най-ранните експерименти в партитурния синтез, той лесно взаймства традиционни форми и използва компютъра, за да прави по-лесен

композиционния процес. Но той работи толкова умело с компютъра, че открива нови величини. Първите три експеримента на Хилър не просто представят отделни пиеси, а цели класове от композиции еквивалентни на избраните форми. Неговият четвърти експеримент генерира нови, непредсказуеми форми, за първи път, които директно променят неговото собствено разбиране за формата. От тези експерименти се появява *“Illiac Suite for String Quartet”* за струнен квартет (1955–1956), от Хилър и Леонард Исаксън (*Leonard M. Isaacson*), която се счита за първата музика, генерирана на дигитален компютър.

Други важни представители на ранния партитурен синтез са Готфрид Кьониг (*Gottfried Koenig*) и Янис Ксенакис (*Iannis Xenakis*). Кьониг разработва композиция, която трябва да бъде изградена стъпка по стъпка: индивидуални величини за различни параметри на композиционната структура; правила за избиране на тези величини и правила за комбинирането им. Ксенакис твърди, че “цялостната детерминирана сложност от действието на композиране представлява една слухова и идеологическа безсмислица”.

СА са дискретни динамични системи, при които пространството, времето и възможностите могат да имат само определен брой състояния. Приложението на музикални алгоритми симулиращи естествен феномен повдига въпроса дали композиторите тайно възприемат алгоритмичното композиране като начин за естествено генериране на естествени форми, които доказват самите себе си чрез собствената си пълнота от естественост. Във всеки случай, моделирането на естествени процеси изисква употребата на компютрите, на които се дължат огромния брой математически изчисления, необходими при моделирането на такива процеси.

В допълнение на основните програмиращи техники и следвайки пълното изпълнение на възможностите на алгоритмичните процеси, композиторият трябва да има дълбоко познание на тяхната математическа формулировка. В днешно време наблюдаваме появата на множество софтуерни програми, които имплементират много алгоритмични процеси. Много от тези програми твърдят, че за да боравиш с тях е необходимо минимално познание по математика. Потребителят трябва само да постави няколко цифри на няколко места и компютърът ще свърши останалото. Въпреки че тези програми могат да бъдат полезни да се изгради груба преценка за алгоритмичните процеси, единственият начин да се разбере процеса е чрез разбирането на неговата математическа формулировка и директното експериментиране с него. Един алгоритъм може да бъде преобразуван по неограничени виртуални начини. Повечето програми

предлагат поне няколко възможности за кодово-графично преобразуване. За да се изследват и наблюдават в пълна степен различни типове кодово-графично преобразуване като математически вариации на един и същ алгоритъм, композитора трябва да имплементира дадения алгоритъм в програмата, използвайки съответно най-подходящия за целта софтуер.

Някои от програмите за алгоритмично композиране се базират върху фракталната геометрия и могат да изчертават миди нотите към конкретни фрактали или фрактални уравнения. Въпреки че подобни програми са широко достъпни и лесни за работа, някои професионални музиканти са им обърнали специално внимание. Музиката, която се създава с такива програми може да наподобява понякога шум или може да звучи съвсем приятно и мелодично. Както при алгоритмичната музика, така и при алгоритмичното изкуство като цяло, всичко зависи от начина, по който параметрите се изчертават към аспектите на тези уравнения, а не от самите уравнения. Поради това, например, същото уравнение може да бъде използвано да създава както лирична и мелодична музика в стила на 19 век, така и фантастични дисонантни какафонии, напомнящи авангардната музика на 50-те и 60-те години на 20 век.

В тази връзка изследователите в областта на хуманните науки осъзнаха през последното десетилетие, че композирането и просвирването на музика от машини е едно само по себе си упражнение в проектирането на дизайна на човешкия интерфейс. Системните дизайнери започнаха да разбират, че колкото по-сложни започнаха да стават компютърните системи, толкова самото наблюдение на състоянието на една такава система ще изисква по-високи и по-обхватни нива на комуникация. Звукът се оказва една очевидна възможност. Изследователите в областта на изкуствения интелект намират музиката за много интересна област, защото тя има пряк досег с човешката активност, както интелектуално, така и емоционално.

Изследователите в областта на телекомуникациите се интересуват от музикалния софтуер, понеже той като една нова форма на предаване на информация, се разпространява много бързо и всеобхватно, а самите изследователи се нуждаят да знаят относно влиянието на основни характеристики на тези форми и как те могат да бъдат манипулирани, следвайки основната цел, тези нови форми да се разпространяват все по-широко.

Философите се интересуват от музикалния софтуер, понеже чрез него са способни да определят музикалното композиране на по-високо ниво, като композирането нота по

нота ще доведе човечеството една стъпка по-близо до директната експресия на музикалните идеи , като по този начин ще даде свобода на човешкия дух.

Софтуерните инженери откриват огромно предизвикателство в области като музикалното композиране; симулацията на толкова сложни човешки дейности изисква опит в алгоритмичния дизайн, експертните системи, оптимизацията и други сходни софтуерно инженерни дисциплини. Да се проектира алгоритъм за композиране на музика, е нещо различно от това да се проектира алгоритъм, който прави инверсия в една матрица. Няма прост, механичен тест за успех. Самият Хилър отбелязва: “Компютърно-базираната композиция е трудна за определение, трудна за ограничаване и трудна за систематизиране.”

Други експерти от тази област изказват подобни мнения: “Всеки опит да се симулират възможностите на човека за композиране най-вероятно няма да има успех, докато музикалните модели, които използват хората са описани и моделирани.”

И напоследък най-често срещаните изказвания са: “Изглежда, че музикалното композиране е трудна, умствена задача, която изисква значително количество опит и усет и всеки сериозен опит да се симулира “некомпютърно” музикално композиране в компютъра ще трябва да се срещне със задачата да изгради един формален модел със значителна сложност. Открихме дори че познанието, което е достигнало алгоритмичното представяне използва хоралния стил на Бах, нещо което вече притеснява будните изследователи. “

Музиката започна да се слуша и създава различно - всичко това благодарение на компютъра и алгоритмичните процедури, които станаха както похват и метод за създаването ѝ, така и принцип на структурална организация на музикалния изказ. Музиката вече говори на много езици използвайки най-универсалния преводач - компютъра.

Както казва Варезе : “Светът се променя и ние се променяме с него. Колкото повече позволяваме на нашето съзнание романтичния разкош да се превръща в съкровище от спомени, толкова по-неспособни ще сме да срещнем бъдещето и да определим новите ценности, които могат да бъдат създадени в него”.

ОСНОВНИ ПРИНОСИ В ДИСЕРТАЦИЯТА

1. Направено е историческо изследване върху развитието, съответно на компютърната, алгоритмична и фрактална музика. Дадени са подробни определения за основните теми, върху които е базирана темата на дисертацията – т.е. определения за компютърна, алгоритмична и фрактална музика. Направен е сериозен исторически обзор на развитието на компютърната и алгоритмичната музика, както и сравнителен анализ върху спецификата на компютърната, електронната и електроакустичната музика. Приведени са сравнителни теми за това как музиката бива превеждана на различни езици – използвайки музикалните закони за структурно съдържание или използвайки протокола MIDI за комуникация между нотите и компютъра.
2. Направен е анализ на новите похвати, които се използват при създаването на компютърна музика, измежду които са дигиталния синтез на музиката и параметричния контрол при синтеза на музиката. Дадени са конкретни определения свързани с понятието компютърна музика. Дадена е важна концепция за създаването на дигитална музика чрез компютърните технологии. Друга важна част от изследователската работа засяга кодово-графичното преобразуване на музика при представяне на многомерни данни и изследване влиянието на различните характеристики при процеса на възприятие.
3. Изведени са интересни факти свързани с историята на фракталите и намирането на естествени взаимовръзки между тяхната структура и принципи на изграждане и тези, на които се основава създаването на музика. Изтъкнати са важни черти на музиката, които имат сходство с фракталността и се прави директна асоциация между музиката, като фрактален модел и самите фрактали. Доказват се и сходни черти между съвременната музика и фракталния модел. Доказва се че немалко явления в музиката проявяват фракталност в структурата си, като едно от основните свойства, които се откриват като сходство между фракталите и музиката са златното сечение, делението на трайността, тактовата организация, лейтмотивните повторения, мотивни и серийни преобразувания.
4. Изследвани са основни методи за създаване на алгоритмична музика, с цел намирането на най-оптималната система в наличните лабораторни условия, за да бъде създаден алгоритъм на музика. Това изследване включва най-известните

методи, които се използват за алгоритмично композиране - стохастични процеси, вероятностни функции, веригите на Марков; дву и триизмерни хаотични системи; фрактали; алгоритми получени от спектрален шум; клетъчна автоматизация, генетични алгоритми, L-системи. Изследват се методи за създаването на фрактална музика, както и специфични черти, обуславящи създаването на тази музика.

5. Изследвани са най-популярните софтуерни продукти, които се използват за създаване на компютърна алгоритмична музика. Създадени са и са приложени музикални примери, създадени с такъв софтуер.
6. С оглед на това, че в дисертацията използвам термина *mapping*, който няма български еквивалент, въвеждам български еквивалент: кодово-графично преобразуване.
7. Това изследване дава предпоставки за създаването на фундамент за изграждането на стройна алгоритмична теория за създаване на универсално музикално произведение.

БИБЛИОГРАФИЯ

1. Бечева, Р. *Електронна музикална интерпретация*. София: Графопринт, 2009.
2. Георгиев, Е. *Музикална акустика*. София: Наука и изкуство, 1975.
3. Детев, Й. *Въведение в музикалната информатика. Компютърните музикални инструменти*. София: 1992.
4. Кавалджиев, Л. *Съвременна музикална терминология*. Българско музикознание, 1994.
5. Лазаров, С. *Електронна музика и синтезатори*. София: Техника, 1986.
6. Лазаров, С., Е., Лазаров. *Компютри и музика*. София: Техника, 1989.
7. Минчева, П., Филева, Кр., *Елементарна теория на музиката*. П., 1996 г.
8. Хаджиев, П. *Елементарна теория на музиката*. София: Музика, 2004.
9. Хаджиев, П. *Хармония*. София: Музика, 1976.
10. Alpern, A. *Techniques for Algorithmic Composition of Music*. Hampshire College, 1995
11. Anderson, J.T. and C.S. Ogilvy. *Excursions in Number Theory*. New York : Dover , 1988. 139-140.
12. Barlow, C. “*Bus Journey to Parametron*”. Feedback Papers, 1980. 21–23.
13. Beauchamp, Horner and Haken. “*Genetic Algorithms and Treir Application to FM Matching Synthesis*”. 17-29.

14. Beyls, P. *"The Musical Universe of Cellular Automata"*. Proceedings of the 1989 International Computer Music Conference. San Francisco: International Computer Music Association, 1989. 34–41.
15. Bidlack, R. *Chaotic Systems as Simple (but Complex) Compositional Algorithms*. Computer Music Journal 16(3) (1992), 33 - 47.
16. Boethius, A. *Fundamentals of Music*. In Strunk's Source Readings in Music History. Ed. O. Strunk. New York: Norton, 1998. 137-143.
17. Boon, J.P. and O. Decroly. *Dynamical Systems Theory for Music Dynamics*. Haos 5(3) (1995), 501-508.
18. Bolognesi, T. *"Automatic Composition: Experiments with Self-similar Music."*. Computer Music Journal, vol. 7, no. 1, 1983. 25–36.
19. Briggs, J. And F. D. Peat. *Turbulent Mirror*. New York: Harper and Row, 1989.
20. Brün, H. *From Musical Ideas to Computers and Back*. In: Harry B. Lincoln. Ed. The Computer and Music Ithaca. Cornell University Press, 1970.
21. Cassiodorus, F. *Fundamentals of Sacred and Secular Learning*. In Strunk's Source Readings in Music History. Ed. O. Strunk. New York: Norton, 1998. 143-148.
22. Clawson, C.C. *Mathematical Mysteries: The Beauty and Magic of Numbers*. New York: Plenum Press, 1966.
23. Darwin, C. *On the Origin of Species*. Cambridge, MA: Harvard University Press, 1964.
24. Dodge, C. and T. A. Jers. *Computer Music: Synthesis, Composition and Performance*. Schirmer Books, 1985.

25. Fournier, A., D. Fussell and L. Carpenter. "*Computer Rendering of Stochastic Models*". Communications of the A.C.M., vol. 25, no. 6, 1982. 371–384.
26. Gardner, M. "*White and Brown Music, Fractal Curves and $1/f$ Fluctuations*". Scientific American 238 (4) (1978): 16-32.
27. Gleick, J. *Chaos*. New York: Penguin Books, 1987.
28. Goldberg, D. and A. Horner, "*Genetic Algorithms and Computer-Assisted Music Composition*". In Proceedings of the 1991 International Computer Music Conference. San Francisco: International Computer Music Association, 1991. 476-482
29. Hiller, L. and L. Isaacson, eds. *Experimental Music; Composing with an Electronic Computer*. New York: McGraw-Hill, 1959.
30. Hiller, L. "*Music Composed with Computers – A Historical Survey*". The Computer and Music, Cornell University Press, 1970. 42–96.
31. Hiller, L. "*Composing with Computers: A Progress Report*". Computer Music Journal, vol. 5, no. 4, 1981. 7–21.
32. Hoffmann, P. "*Zelluläre Automaten als kompositorische Modelle. Sind Chaos und Komplexität musikalische Phänomene?*" In Arbeitsprozesse in Physik und Musik. Frankfurt am Main/Berlin: Peter Lang Verlag/Akademie der Künste Berlin, 1994. 7–18.
33. Hunt, A., R. Kirk, and R. Orton. "*Musical Applications of a Cellular Automata Workstation*". In Proceedings of the 1991 International Computer Music Conference. San Francisco: International Computer Music Association, 1991. 165-168.
34. James, J. *The Music of the Spheres: Music, Science and the Natural Order of the Universe*. London: Abacus, 1993.
35. Jacob, Br. *Algorithmic Composition as a Model of Creativity*. Organised Sound, vol. 1, No3, 1996.

36. Jerez, Gustavo-Diaz. *Algorithmic music: using mathematical models in music composition*. Doctoral dissertation, The Manhattan School of Music. 2000.
37. Jones, K. “*Compositional Applications of Stochastic Processes*”. *Computer Music Journal* 5(2) 1981 : 45 - 61.
38. Knuth, D. E. “*The Complexity of Songs*”. *SIGACT News* vol. 9, no. 2, 1977. 17–24.
39. Koza, L.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press, 1992.
40. Koenig, G. M. *Ästhetische Praxis. Texte zur Musik*. Band 3: 1968–1991. Saarbrücken: Pfau.
41. Kurepa, A. and R. Waschka, “*Using Fractals in Timbre Construction: an Exploratory Study*”. In *Proceedings of the 1989 International Computer Music Conference*. San Francisco: International Computer Music Association, 1989. 332-335.
42. La Motte-Haber, H. de. “*Alles ist Zahl. Formen pythagoreischen Denkens in der Musik*”. In T. Ott, and H. von Loesch, eds. *Musik befragt—Musikvermittelt*, Augsburg: Wißner, 1996. 153–163.
43. La Motte-Haber, H. de. “*Struktur und Programm. Analytische Betrachtungen zur Komposition Summa von Arvo Pärt*”. In W. Gratzner, ed. *Nähe und Distanz. Nachgedachte Musik der Gegenwart 1*. Hofheim/Ts.: Wolke, 1996. 14-25.
44. Langston, P. S. “*The Influence of Unix on the Development of Two Video Games*”. *EUUG Spring '85 Conference Proceedings*, 1985.
45. Langston, P. S. *An Experiment in Music Generation*. *Usenix Summer '86 Conference Proceedings*, 1986.

46. Levine, D. and P.S. Langston, *Video game for the Atari 800, & Commodore 64 home computers*. 1984.
47. Lindenmayer, D. *Mathematical Models for Cellular Interactions in Development*, Parts I-II., *Journal of Theoretical Biology* 18 (1968) : 280-315
48. Lorenz, E. N. *Deterministic Nonperiodic Flow*. *Journal of the Atmospheric Sciences* 20 (1963): 130-141.
49. Loy, G. “*Composing with computers*”. A Survey of some Compositional Formalisms and Music Programming Languages. In *Current Directions in Computer Music Research*. Eds. M. Mathews and J. R. Pierce. Cambridge: MIT Press, 1989. 292 - 396.
50. Mandelbrot, B. *The Fractal Geometry of Nature*. San Francisco: W.H. Freeman, 1982.
51. McNabb, M. “*Dreamsong: The Composition*”. *Computer Music Journal*, vol. 5, no. 4, 1981. 36–54.
52. Mason, S. and M.Saffle. “*L-Systems, Melodies and Musical Structure*”. *Leonardo Music Journal* 4 (1994): 53-58.
53. Maurer IV, A. *A Brief History of Algorithmic Composition*. 1999
54. “*MIDI 1.0 Detailed Specification*”. The International MIDI Association, 11857 Hartsook St., No. Hollywood, CA 91607, (818) 505-8964, (1985).
55. Miranda, E. “*Cellular Automata Music: An Interdisciplinary Project*”. *Interface* 22 (1993): 3-21.
56. Miranda, E. “*Music Composition Using Cellular Automata*”. *Languages of Design* 2 (1994) : 105-117.

57. Moorer, J. A. “*Music and Computer Composition*”. *Communications of the ACM*, vol.15, no. 2, 1972. 104.
58. *New Harvard Dictionary of Music*. Ed. Don Randel. S. V. “Pythagorean hammers”. Cambridge, MA : Harvard Univ. Press, 1986.
59. Ott, E. *Chaos in Dynamical Systems*. New York: Cambridge University Press, 1993.
60. Peitgen, H.O. and D.Saupe, eds. *The Science of Fractal Images*. New York: Springer, 1988.
61. Pickover, C.A. *Million – Point Sculptures*. *Computer Graphics Forum* 10(4) (1991) : 333-336.
62. Potter, G. M. “The Role of Chance in Contemporary Music”. Doctoral dissertation, Indiana University, University Microfilms, 1971.
63. Prusinkiewicz, P. “*Score Generation with L-Systems*.” *Proceedings of the 1986 International Computer Music Conference*. San Francisco: International Computer Music Association, 1986. 455–457.
64. Prusinkiewicz, P., and A. Lindenmayer. *The Algorithmic Beauty of Plants*. New York: Springer-Verlag, 1990.
65. Redman, B. E. “*A User Programmable Telephone Switch*”. Bellcore Technical Memorandum, 1987.
66. Schillinger, J. *The Schillinger System of Musical Composition*. New York: Da Capo Press, 1978.
67. Schillinger, J. *The Mathematical Basis of the Arts*. New York: The Philosophical Library, 1948.

68. Smith, E. Liner notes to W.A.Mozart, *A Musical dice game*. Phillips Complete Mozart Edition-Rarities and Surprises 422 542 -2, 1991.

69. Smith, A. R. “*Plants, Fractals, and Formal Languages*”. Computer Graphics Proceedings of the Siggraph '84 Conference, vol. 18, no. 3, 1984. 1–10.

70. “*Structure and function in musical repetition*”. Journal of the Canadian association of University School of Music, 1978.

71. “*The element of repetition in nature and the arts*”. Musical Quarterly, 12/1963.

72. Varese, E. *The Liberation of Sound*. In Strunk’s Source Readings in Music History. Ed. O. Strunk. New York: Norton, 1998. 1339 - 1346

73. Vetterling, W.T. Ed. *Numerical Recipes in C*. New York: Cambridge Univ. Press, 1997.

74. Von Neuman, J. *Theory of Self-Reproducing Automata*. Champaign, IL: University of Illinois Press, 1966.

75. Voss, R. F. and J. Clarke. “*1/f noise in music and speech*”. Nature 258 (1975) : 317-318.

76. VOSS78 Voss, R. F. and J. Clarke. “*1/f Noise in Music: Music from 1/f Noise*”. Journal of the Acoustical Society of America, vol. 63, no. 1, 1978. 258–263.

77. Wolfram, S. ed. *Theory and Applications of Cellular Automata*. Singapore: World Scientific, 1986.

78. Xenakis, Iannis. *Formalized Music; Thought and Mathematics in Music*. New York: Pendragon Press, 1992.

79. Zarlino, G. *Institutioni harmoniche*. Trans. G. Tomilson. Ridgewood, N. J. : Gregg Press, 1966.

ИНТЕРНЕТ ИСТОЧНИЦИ

80. http://muse.jhu.edu/journals/computer_music_journal/v025/25.1supper.html
81. <http://www.langston.com/Papers/amc.pdf>
82. <http://homepages.rpi.edu/~eglash/eglash.htm>
83. <http://www.ccd.rpi.edu/Eglash/csdt/african.html>
84. http://www.ccd.rpi.edu/Eglash/csdt/african/African_Fractals/homepage.html
85. <http://strangepaths.com/canon-1-a-2/2009/01/18/en/>
86. <http://www2.nau.edu/tas3/musoffcanons.html>
87. <http://www2.nau.edu/tas3/mirrorfuguesaof.html>
88. http://www.flagmusic.com/aof.php?r=aof_c_xii
89. <http://www.teoria.com/articulos/analysis/kdf/XII/eng/>
90. http://www.flagmusic.com/aof.php?r=aof_c_xiii
91. http://www.flagmusic.com/aof.php?r=aof_c_c1
92. <http://ccrma-www.stanford.edu/~blackrse/algorithm.html>]
93. http://www.ee.umd.edu/~blj/algorithmic_composition/algorithmicmodel.html
94. <http://www.geocities.com/Athens/Academy/8764/lmuse/lmuse.html>
95. http://www.gustavodiazjerez.com/fractmus_algorithm.html#Wolfram CA

96. <http://www.musoft-bulders.com>
97. www.fractalcomposer.com
98. <http://www.fractalmusiclab.com>
99. www.forwiss.uni-erlangen.de/~kinderma/musinum
100. <http://www.musoft-builders.com>
101. <http://www.maths.gla.ac.uk/~km/dsysmus.htm>
102. <http://www.organised-chaos.com/oc/ginger.html>
103. <http://www.fin.ne.jp/~yokubota/>
104. <http://www.cisnet.or.jp/home/magari>
105. <http://symcom.hypermart.net>.
106. <http://mitpress.mit.edu/e-books/csound/fpage/getCs/getcs.html>
107. <http://members.aol.com/dsinger594/caman>
108. <http://www.randomtunes.com>
109. http://en.wikipedia.org/wiki/Mandelbrot_set
110. <http://www2.vo.lu/homepages/armand/index.html>
111. <http://autoinfo.smartlink.net/ray>
112. <http://www-ks.rus.uni-stuttgart.de/people/schulz/fmusic/wtf>

113. <http://www.geocities.com/Athens/Academy/8764/lmuse/lmuse.html>
114. <http://www.algoart.com/software/index.htm>
115. <http://www.engr.newpaltz.edu/~chan12/phase.html>
116. <http://www.fractal-wallpapers.com/Fractal-Wallpapers/Fractal-Art-Wallpaper/>
117. <http://www.koncon.nl/ACToolbox>
118. <http://www-ccrma.stanford.edu/CCRMA/Software/cm/cm.html>
119. <http://thompsonresidence.com/keykit>
120. <http://www.essl.at/works/rtc.html>
121. <http://www.softsynth.com/hmsl>
122. <http://cycling74.com/products/maxmspjitter/>
123. <http://music.columbia.edu/cmix>
124. http://members.value.com.au/christie/rub_full.html
125. <http://members3.jcom.home.ne.jp/strangemusic/composee.html>
126. <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/music/web/music.software.html>
127. <http://www.youtube.com/watch?v=7Cn7ZW8ts3Y>
128. <http://www.youtube.com/watch?v=LLbpQl1cCl8>
129. http://www.ssm.uk.net/Schillinger_Theory.pdf
130. <http://www.goodreads.com/book/show/5995797-electronic-and-experimental-music-3rd-edition.pdf>

131. <http://duskblue.org/proj/toymidi/midiformat.pdf>
132. <http://www7.scu.edu.au/1841/com1841.htm>
133. <https://jyx.jyu.fi/dspace/bitstream/handle/123456789/10031/154.pdf?sequence=1>
134. <http://vlab.infotech.monash.edu.au/tutorials/cellular-automata/>
135. <http://solomonsmusic.net/misc.htm>
136. <http://aix1.uottawa.ca/~jkhoury/markov.htm>
137. <http://www.intermorphic.com/tools/noatikl/index.html>
138. http://en.wikipedia.org/wiki/H%C3%A9non_map
139. <http://scratch.mit.edu/galleries/view/59341>
140. <http://www.fraktalwelt.de/myhome/simpiter2.htm>
141. <http://www.gustavodiazjerez.com/fractmus/algorithm.html#Henon>
142. [http://www.gustavodiazjerez.com/fractmus/algorithm.html#Gingerbread man](http://www.gustavodiazjerez.com/fractmus/algorithm.html#Gingerbread%20man)
143. <http://www.gustavodiazjerez.com/fractmus/algorithm.html#Martin>
144. http://en.wikipedia.org/wiki/Clifford_A._Pickover
145. http://en.wikipedia.org/wiki/R%C3%B6ssler_attractor
146. http://www.ee.umd.edu/~blj/algorithmic_composition/icmc.95.html
147. <http://bg.wikipedia.org/wiki/KochFlake.png>

148. http://www.ee.umd.edu/~blj/algorithmic_composition/algorithmicmodel.html
149. <http://www.hitsquad.com/smm/programs/TEXTURE/>
150. <http://www.reglos.de/musinum/>
151. http://www.hitsquad.com/smm/programs/Amusical_Generator/
152. <http://www.scribd.com/doc/57779746/87/Brain-interface-IBVA-system>
153. http://www.fractovia.org/art/fractal_generators/music01.shtml
154. <http://en.wikipedia.org/wiki/Max/MSP>
155. <http://www.musicmasterworks.com/WhereMathMeetsMusic.html>
156. http://en.wikipedia.org/wiki/Chromatic_scale
157. http://en.wikipedia.org/wiki/Chromatic_circle
158. <http://divinecosmos.e-puzzle.ru/2Chapter13.htm>
159. http://en.wikipedia.org/wiki/Equal_temperament
160. http://en.wikipedia.org/wiki/Golden_ratio
161. <http://www.goldennumber.net/acoustics.htm>
162. http://en.wikipedia.org/wiki/The_Musical_Offering
163. <http://www.worldofescher.com/gallery/A29.html>
164. http://en.wikipedia.org/wiki/The_Art_of_Fugue

ПРИЛОЖЕНИЕ 1

АНАЛИЗ ВЪРХУ РАЗВИТИЕТО НА МУЗИКАТА, ГЕНЕРИРАНА И КОМПОЗИРАНА С ПОМОЩТА НА МАТЕМАТИКАТА

МУЗИКАТА И МАТЕМАТИКАТА

Музиката и математиката имат хилядолетна взаимосвързана история. Още в класическа древна Гърция възникнат много учения за хармонията, от които най-дълбока следа в световната култура оставя Питагорейското учение. Последователите на гръцкия математик Питагор (560-480 пр.н.е.) си представят Вселената, Космоса и Човека като единно цяло, където всичко е свързано и е в хармонични взаимоотношения. Те представят хармонията като начало на реда, като сила, побеждаваща хаоса. Хармонията е присъща на природата и изкуството: "Едни и същи закони съществуват за музикалните гами и за планетите".

Поради тези представи в античността музиката се счита за подразделение на математиката. Флавиус Касиодорус (*Flavius Cassiodorus* 485-575) е човекът изиграл важна роля в разпространението на античната култура в латинските средни векове, който описва математиката като обединение на четири дисциплини: аритметика, музика, геометрия и астрономия. Математическата наука е тази която засяга абстрактните величини и има следните подразделения: аритметика, музика, геометрия и астрономия.⁸² Музиката е дисциплината, която има отношение към числата и тяхната връзка към онези неща, които се откриват в звуците.[21]

Както се разбира от Глава 2 през цялата история на музиката, много композитори използват математически модели като източник на композиционно творчество. Питагор твърди, че всички неща са числа или че всички тези неща имат отношение към числата. Едното произлиза от другото.[58] И по този начин тт древни времена всички учени и мислители стигат да едно унифицирано заключение и то е, че изкуствата и евентуално всяка интелектуална дейност, са погълнати от числата. Но често се забравя, че в музиката се съдържа голяма част абсорбирана доза от математиката. В теорията на Питагор, числата са основна част от музикалното преживяване, като мистерията на

⁸² F.Cassiodorus, *Fundamentals of Sacred and Secular Learning*, in Strunk's *Source Readings in Music History*, ed.O.Strunk (New York:Norton,1998), 144-145

хармонията е решена и проектирана благодарение на универсалните числови коефициенти. [34]

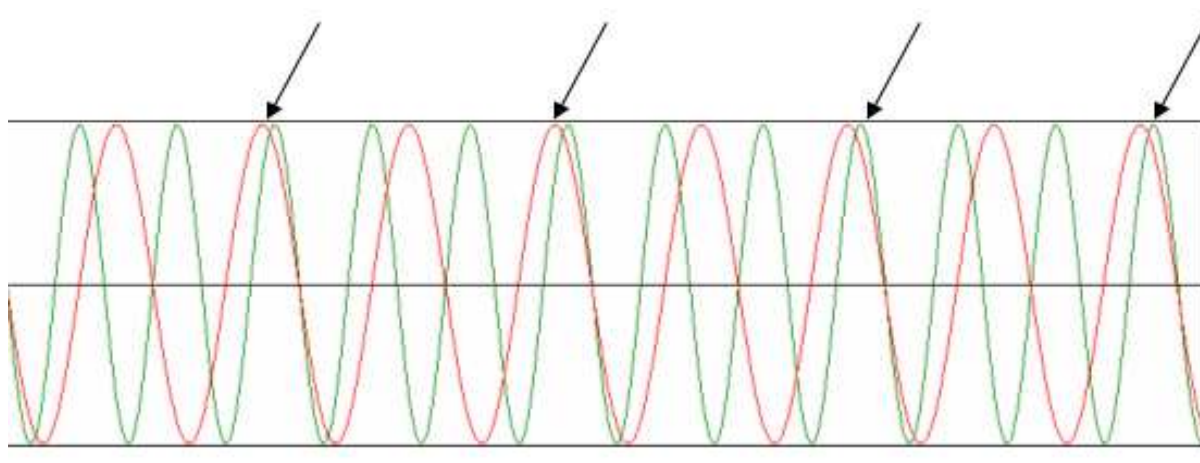
Затова в изкуството се наблюдава строга закономерност на взаимните преминавания на хармоничните и дисхармонични стилове, на лявополукълбовите и дяснополукълбовите тенденции, на Аполоновото и Дионисиевото начало, или по-общо, на Космоса и хаоса. Един много известен пример за връзката между музиката и математиката е настройката на камертона и тона ла с честота на звучене = **440 Hz**.

ЗВУК, ХАРАКТЕРИСТИКИ НА ЗВУКА

Настоящото приложение се фокусира над възможностите на звука и някои от неговите характеристики, които могат да допринесат за по-добро възприемане на заобикалящата среда. Както е известно характеристиките на звука са четири – сила, трайност, тембър и височина. Оказва се, че зад всяка една от тях стои определена числова закономерност.

В тази връзка е интересна абсолютната височина на звука, с оглед предмета на изследване. Абсолютната височина варира през годините. Например в Англия през 1720 г. тонът ла от първа октава е с честота 380 Hz, докато органите, на които е свирил Йохан Себастиан Бах (*Johann Sebastian Bach*) в Хамбург, Ваймар и Лайпциг са били с честота $A = 480$ Hz. През 1939, на Международна конференция са приема, че A (Ла) от първа октава се фиксира с абсолютна височина от 440 Hz, известна още като концертна височина - *concert pitch*. [8] Това се налага като стандарт за строй в почти всички световни оркестри. Но например Бостънската филхармония използва строй: $A = 442$, а строй от $A = 445$ Hz може да бъде чул в Германия, Австрия и Китай.

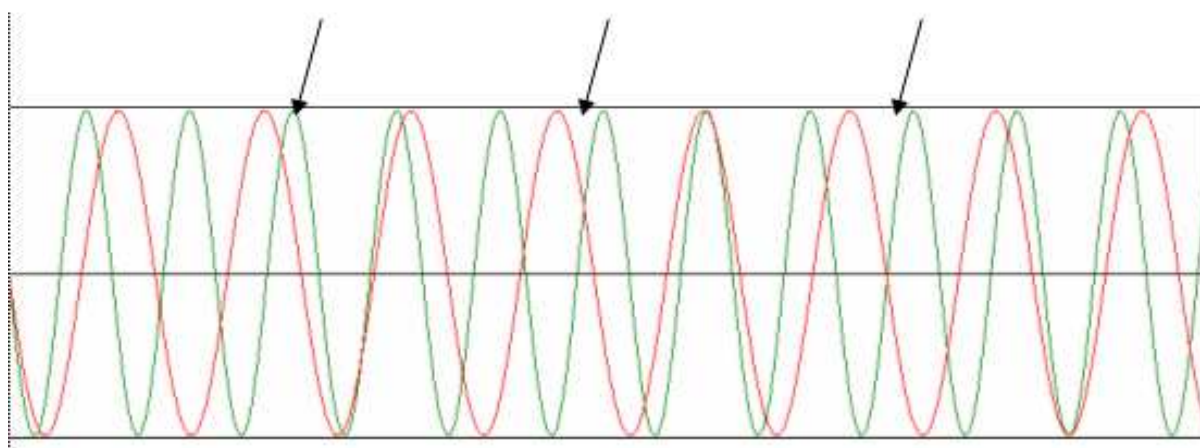
Също така погледнато от математическа гледна точка, можем да си обясним явленията консонанс и дисонанс. За да се даде логично обяснение защо някои интервали звучат добре, а други не толкова добре, трябва първо да се разбере строежа на звуковата вълна, създадена от един инструмент. Когато се дръпне струната на даден струнен инструмент, тя вибрира. Това предизвиква механична енергия, която предава тази вибрация по въздуха, в звукови вълни. Нека да разгледаме звуковите вълни на 2 тона, които звучат добре заедно (Фиг.1). Използваме тоновете C (До) и G (Сол) от първа октава:



Фигура 1. Звукови вълни на тоновете С и G от първа октава [155]

На графиката има повтарящ се образец: всяка 3-та вълна или трептение на G (зелен цвят) съвпада с всяка 2-ра вълна на С. Това е тайната за създаване на добре звучащи тонови комбинации: честоти, които съвпадат на равномерни интервали.

Нека сега разгледаме два тона, С (До) and F# (Фа#) (Фиг. 2), които не звучат добре едновременно:



Фигура 2. Звукови вълни на тоновете С и F# от първа октава [155]

В тази графика няма такива съвпадения. Първият интервал звучи ‘консонантно’, а вторият ‘дисонантно’.

Чистата квинта е интервал между два тона, отстоящи на 7 полутона един от друг. Съотношението на перфектната квинта е 3:2 (която разгледахме в първата графика), което означава, че по-високия тон прави три вибрации за единица време, докато по-ниския тон прави две вибрации.

Нека сега разгледаме едно тризвучие, за да разберем защо звучи добре . Честотите на тоновете от това тризвучие са върху акорда⁸³ До мажор (до от първа октава):

C – 261.6 Hz

E – 329.6 Hz

G – 392.0 Hz

Отношението на E (Ми) към C (До) е около $5/4$. Това означава, че всяка 5-та вълна на E (Ми) съвпада с всяка 4-та вълна на C (До). Отношението на G (Сол) към E (Ми) е също около $5/4$. Отношението между G (Сол) и C (До) е около $3/2$. Докато честотата на всеки тон съвпада на постоянни интервали с честотата на другия тон - всички звучат добре! Това е един елементарен пример за математическото присъствие в един интервал.

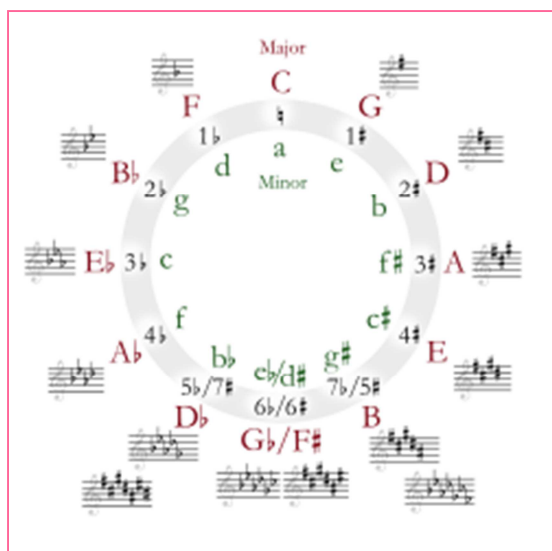
Интересна връзка може да се направи и между алгоритъмът на атоналната музика, която има своите особености, залегнали в създаването на музикалния фрактал. Същото се отнася и за додекафоничната музика, чиято поредица от 12 полутона може да бъде подложена на сериозна математическа обработка. (Фиг. 3)



Фигура 3. Хроматична гама [156]

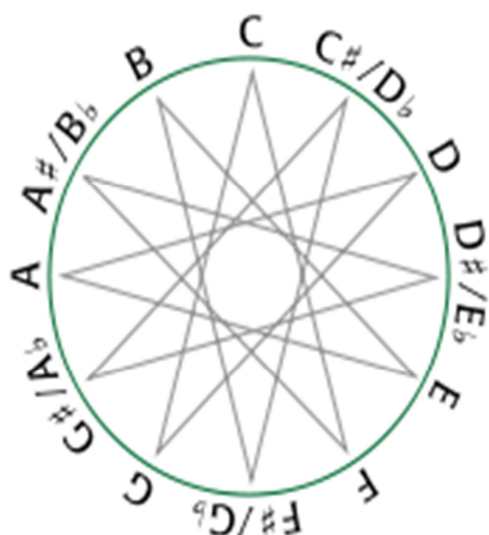
От математическа гледна точка е интересен и хроматичният кръг, показващ връзката между 12-те равномерно темперирани части от хроматичната скала. (Фиг. 4) Ако се вземе който и да е тон от тази скала и започне да нараства постоянно с полутон, със сигурност ще съвпадне височинно в един момент с височината на първоначалния тон, преминавайки през всички останали равномерно темперирани части.

⁸³ Акорд е едновременно звучене (т.е. съзвучие) на три или повече различни по височина тона, които са наредени, или могат да се наредят, по терци /7.с.27/



Фигура 4. Хроматичен квинтов и квартов кръг [157]

Кръгът на квинтите в хроматичната скала може да се илюстрира с дванадесетоъгълник. (Фиг. 5)



Фигура 5. Хроматичната скала в геометричното пространство [157]

От гледна точка на алгоритмизирането на музиката интересна характеристика е и ритъма - организираното, логично редуване на тонови трайности. Интересно е че в човешкия организъм множество биологични процеси протичат в съответствие с някакъв вътрешен ритъм. Най-ясно отличим е ритъмът на сърцето, който се забързва и забавя в зависимост от физическото и емоционалното състояние на човек. Ако за нормален сърдечен ритъм се смятат от 70 до 80 удара, то баладичните мелодии варират в темпо между 60 и 90 удара, докато бързите мелодии са в темпо над 100 удара.

МАТЕМАТИЧЕСКИ ЗАВИСИМОСТИ В МУЗИКАЛНИЯ СТРОЙ

Една от известните дефиниции за музикален строй е тази, която представя музикалния строй, като определен ред от звукове с характерна зависимост между тях. Относителната височина на тоновете в една скала може да бъде определена чрез даден музикален строй.

Както стана известно малко по-рано, в 6 век пр.н.е. първият, цялостно разработен математическият строй е Питагоровия строй, който още е прието да се нарича строй на чистите квинти и кварта. При него полутоновият интервал се определя с отношението 16:15.

ПИТАГОРОВ СТРОЙ

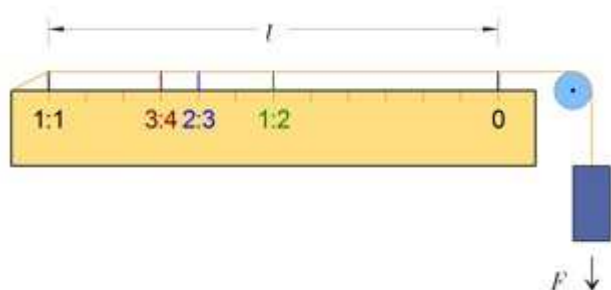
За Питагор музиката е била начин за физическото изразяване на математиката. Също така музиката в тези времена е била част от четирите изкуства в *Quadrivium* – геометрия, алгебра, музика и астрономия. Изследвайки монохорда (еднострунен музикален инструмент в Древна Гърция) през 530 г.пр.н.е. Питагор достига до извода че $\frac{2}{3}$ от дължината на една струна възпроизвежда тон с чиста квинта по висок от основния тон на цялата струна (Фиг.6). Така той създава поредица от тонове подредени по квинтовия кръг. Питагор създава първата музикална система, годна да записва музикалните композиции и да ги препредава във времето и пространството. Това откритие е толкова значимо за тогавашния свят, че става основа на теорията за "Музика на сферите".⁸⁴ Твърденията на древногръцкия учен Питагор са, че Земята е кълбо и, че Слънцето, Луната, звездите и петте видими планети (Меркурий, Венера, Марс, Юпитер и Сатурн) се въртят около нея по осем концентрични сфери, без да си влияят помежду си. Търкането на сферите създава хармонични, неуловими за ухото звуци, наречени "музика на сферите". Хармонията на сферите е важно понятие през цялото време от Питагор до Кеплеровата Космография през 1596 г. Едва 1500 години след Питагор, в средните векове, композиторите започват да формулират правилата, чрез които височинните връзки и комбинации биват определени.

⁸⁴ A.Boethius, Fundamentals of Music, in Strunk's Source Readings in Music History, ed.O. Strunk (New York:Norton,1998), 140



Фигура 6. Монохорд - еднострунен инструмент [158]

Използването на монохорда като експериментален инструмент от музиканти и теоретици, като Андреас Веркмайстер (1645-1706) и Йохан Найдхарт (1685-1735), показва, че интереса за връзката между математика и музика датира от древни времена. Питагоровата интонационна система е настройка, която се базира на серия от перфектни квинти. Питагоровият строй оптимизира квинтите (отношение 3:2) и техните обръщания, квартите (отношение 4:3), правейки възможно да се получават тези интервали в техните най-хармонични форми. Съобразно тези отношения се дефинира основния звук на готическата полифония. Питагоровият строй се основава на голям брой интервали, всеки от които е настроен в съотношение 3:2, следващото най-просто съотношение след 2:1 (Фиг.7). Този избор не е само математически елегантен, но също така е и най-лесния за възприемане от човешкия слух.



Фигура 7. Деление на струна в съотношения [158]

За да се изчисли честотата на тон в скалата от съотношения, честотното отношение се умножава по тоническата (основна) честота. За пример, чрез тоника А (Ла) с честота = 440 Hz, може да се построи квинта над А (Ла), която ще бъде $440 \cdot (3:2) = 660$ Hz. Започвайки от тона D (Ре), се възпроизвеждат шест други тона, чрез

движение шест пъти нагоре в съотношение 3:2 и придвижвайки оставащите тонове в същото отношение надолу.

Eb—Bb—F—C—G—[D]—A—E—B—F#—C#—G#

Ако A(Ла) над D(Ре) е настроена така, че честотното отношение на A(Ла) към D(Ре) е 3:2, тогава ако D(Ре) е настроено на 288 Hz, а A(Ла) е настроено на 432 Hz. E(Ми) над A(Ла) също е в отношение 3:2, което поставя E(Ми) на 648 Hz.

В темперирания строй двойките енхармоници, като Eb (Миб) и D# (Ре#) се възприемат като еднакво звучащи, докато в Питагоровия строй, те теоретично имат различни съотношения и имат различна честота. Това несъответствие от около 23.5 cents, или една четвърт от полутон, е известно като Питагоров интервал - *Pythagorean comma*. [58]

За да се справи с този проблем Питагоровият строй използва 12 тона от Eb (Миб) до D# (Ре#), след което поставя над D# (Ре#) още едно Eb (Миб), започвайки последователността отново. Това прави интервала D# (Ре#)—Eb (Миб) да звучи зле – извън строя, което означава, че всяка музика, която комбинира тези два тона е неупотребима в този строй. Много интервали извън строя като този са познати като дисхармония или вълчи интервал (*wolf interval*).

Много от тези правила впоследствие залягат в основата на фракталната музика.

ТОЧНО ТЕМПЕРИРАН СТРОЙ (JUST INTONATION)

В музиката, точен (правилен) е всеки музикален строй, в който честотите на тоновете са свързани чрез съотношения на цели числа. По този начин се построяват най-добре звучащите за слуха интервали, като квинта, кварта и терца. Те се изписват или като съотношения с дуоточие (3:2), или като обикновена дроб (3/2). Точно темперираният строй е конструиран на базата на интервала октава и се опитва да съчетава колкото се може повече от тези добре звучащи интервали. Всеки интервал настроен по този начин се нарича точен интервал (*just interval*) или с други думи двата тона са част от същата хармонична серия. При точно темперирания строй полутоните не са разположени на еднакво разстояние.

За съжаление, точно темперираният строй зависи от тоналността, която използваме - строят за До мажор не е същия като при Ре мажор. Точно темперираният

строй се използва за ансамбли, изпълняващи хорали или оркестрови творби, при които изпълнителите съчетават строя един към друг слухово.

РАВНОМЕРНО-ТЕМПЕРИРАН СТРОЙ (12 TET – TONE EQUAL TEMPERAMENT)

Добър пример за връзката между музиката и математиката е равномерно-темперирания (съвременния) строй⁸⁵, който се характеризира с това, че при него чистата октава запазва съотношението $\frac{1}{2}$, но се дели на 12 равни полутона. При така създадената температура отношението на честотите между два съседни полутона е постоянна величина, т.е. височинните стойности в този строй образуват геометрична прогресия с коефициент

$$k = \sqrt[12]{2} \approx 1,059463$$

Полутонът е най-малкото разстояние в съвременния темпериран строй. Такива строеве се измерват в т.нар. *центове*⁸⁶, като един полутон (най-малкото разстояние между тоновете е равен на 100 цента), докато в една октава има 1200 Cents.

Най-важното при равномерното темперирание е че при него от всеки произволно взет тон чрез умножение или деление на k може да се получи всеки друг тон. Неговата 12-тонова система се характеризира с общоприети абсолютни стойности на честотите на тоновете. Опорен тон за целия строй е тонът “ла” с честота 440 Hz.[5]

В таблица 1 са изведени честотите на тоновете съвременния темпериран строй. Това разпределение на честотите е прегледно за настоящия анализ.

⁸⁵ Равномерно-темпериран строй е европейската диатонична система, при която октавата е разделена на 12 полутона, които могат теоретично да се подредят в 12-степенния квартов-квинтов кръг./1, с.112/

⁸⁶ Системата, която английският музикален акустик Александър Джон Елис е изобретил за измерване на тоновата височина е т.нар.центова система. Тя разделя темперования полутон на 100 равни части /1, с.114/

**Таблица с честотите на тоновете , определени спрямо А4 = 440 Нз , а
С от 1 октава е С4**

Честота в херци										
Октава → Тон ↓	0	1	2	3	4	5	6	7	8	9
C	16.352 (-48)	32.703 (-36)	65.406 (-24)	130.81 (-12)	261.63 (0)	523.25 (+12)	1046.5 (+24)	2093.0 (+36)	4186.0 (+48)	8372.0 (+60)
C#/Db	17.324 (-47)	34.648 (-35)	69.296 (-23)	138.59 (-11)	277.18 (+1)	554.37 (+13)	1108.7 (+25)	2217.5 (+37)	4434.9 (+49)	8869.8 (+61)
D	18.354 (-46)	36.708 (-34)	73.416 (-22)	146.83 (-10)	293.66 (+2)	587.33 (+14)	1174.7 (+26)	2349.3 (+38)	4698.6 (+50)	9397.3 (+62)
Eb/D#	19.445 (-45)	38.891 (-33)	77.782 (-21)	155.56 (-9)	311.13 (+3)	622.25 (+15)	1244.5 (+27)	2489.0 (+39)	4978.0 (+51)	9956.1 (+63)
E	20.602 (-44)	41.203 (-32)	82.407 (-20)	164.81 (-8)	329.63 (+4)	659.26 (+16)	1318.5 (+28)	2637.0 (+40)	5274.0 (+52)	10548 (+64)
F	21.827 (-43)	43.654 (-31)	87.307 (-19)	174.61 (-7)	349.23 (+5)	698.46 (+17)	1396.9 (+29)	2793.8 (+41)	5587.7 (+53)	11175 (+65)
F#/Gb	23.125 (-42)	46.249 (-30)	92.499 (-18)	185.00 (-6)	369.99 (+6)	739.99 (+18)	1480.0 (+30)	2960.0 (+42)	5919.9 (+54)	11840 (+66)
G	24.500 (-41)	48.999 (-29)	97.999 (-17)	196.00 (-5)	392.00 (+7)	783.99 (+19)	1568.0 (+31)	3136.0 (+43)	6271.9 (+55)	12544 (+67)
Ab/G#	25.957 (-40)	51.913 (-28)	103.83 (-16)	207.65 (-4)	415.30 (+8)	830.61 (+20)	1661.2 (+32)	3322.4 (+44)	6644.9 (+56)	13290 (+68)
A	27.500 (-39)	55.000 (-27)	110.00 (-15)	220.00 (-3)	440.00 (+9)	880.00 (+21)	1760.0 (+33)	3520.0 (+45)	7040.0 (+57)	14080 (+69)
Bb/A#	29.135 (-38)	58.270 (-26)	116.54 (-14)	233.08 (-2)	466.16 (+10)	932.33 (+22)	1864.7 (+34)	3729.3 (+46)	7458.6 (+58)	14917 (+70)
B	30.868 (-37)	61.735 (-25)	123.47 (-13)	246.94 (-1)	493.88 (+11)	987.77 (+23)	1975.5 (+35)	3951.1 (+47)	7902.1 (+59)	15804 (+71)

Таблица1. Честотите на тоновете в равномерно- темперирания строй [5]

Голяма заслуга за бързото възприемане на “темперираната система” има Й.С.Бах чрез своите композиции за “Добре темперирано пиано”, включващи 24 прелюдии и фуги във всички мажорни и минорни тоналности, за да може всяка тоналност да звучи добре от който и да е натиснат клавиш. Пълното наименование на цикъла е “Добре темперирано пиано, или прелюдии и фуги във всички тонове и полутонове, касаещи както мажорните, така и минорните тоналности”.

Темперираният строй е използван успешно в музиката от 19 и 20 век, в творбите на композиторите Шуман, Брамс, Чайковски и Дебюси. Създаден е за клавишните инструменти, за да може всеки който композира или свири на тях, да го прави от който и да е клавиш на пианото и това да звучи еднакво добре, независимо от избора на клавиш.

Не спират опитите да се създадат и наложат и други равно-темперирани строеве. Съществуват и други равномерни темперации, като някои творци пишат музика в 19-ТЕТ и 31-ТЕТ, а арабската музика се основава на 24-ТЕТ.

СРАВНИТЕЛЕН АНАЛИЗ НА РАЗЛИЧНИТЕ СТРОЕВЕ

За да получим алгоритмичен модел на универсалната формула за създаване на фрактал е необходимо да изследваме приликите и разликите между различните строеве. Това е направено в табличен вид. (Таблицы 2 и 3)

СРАВНЕНИЕ МЕЖДУ ПИТАГОРОВ СТРОЙ И РАВНОМЕРНО-ТЕМПЕРИРАН СТРОЙ

Тон	Интервал	Съотношения в Питагоров строй	Съотношения в 12 ТЕТ
C	Унисон	1.0000	1.0000
Db	Малка секунда	$256/243 = 1.0535$	1.05946
D	Голяма секунда	$9/8 = 1.1250$	1.12246
Eb	Малка терца	$32/27 = 1.1852$	1.18921
E	Голяма терца	$81/64 = 1.2656$	1.25992
F	Кварта	$4/3 = 1.3333$	1.33484
F#	Умалена квинта	$729/512 = 1.4238$	1.41421
G	Квинта	$3/2 = 1.5000$	1.49831
Ab	Малка секста	$128/81 = 1.5802$	1.58740
A	Голяма секста	$27/16 = 1.6875$	1.68179
Bb	Малка септима	$16/9 = 1.7778$	1.78180
B	Голяма септима	$243/128 = 1.8984$	1.88775
C	Октава	2.0000	2.0000

Таблица 2. Сравнение между питагоров и равномерно-темпераан строй [159]

СРАВНЕНИЕ МЕЖДУ ТОЧНО ТЕМПЕРИРАН СТРОЙ И РАВНОМЕРНО –
ТЕМПЕРИРАН СТРОЙ

Тон	Интервал	Съотношения в Точно темперирани строй	Съотношения в Равномерно- темперирани строй
C	Унисон	1.0000	1.0000
Db	Малка секунда	$16/15 = 1.0667$	1.05946
D	Голяма секунда	$9/8 = 1.1250$	1.12246
Eb	Малка терца	$6/5 = 1.2000$	1.18921
E	Голяма терца	$5/4 = 1.2500$	1.25992
F	Кварта	$4/3 = 1.3333$	1.33484
F#	Умалена квинта	$45/32 = 1.4063$	1.41421
G	Квинта	$3/2 = 1.5000$	1.49831
Ab	Малка секста	$8/5 = 1.6000$	1.58740
A	Голяма секста	$5/3 = 1.6667$	1.68179
Bb	Малка септима	$16/9 = 1.7778$	1.78180
B	Голяма септима	$15/8 = 1.8750$	1.88775
C	Октава	2.0000	2.0000

Таблица 3. Сравнение между точно темперирани строй и равномерно-темперирани строй [159]

В таблицата по-долу (Таблица 4) разликата между съотношенията в точно темперирания и равномерно-темперирания строй е изразена в центове. Интервалите от равномерно-темперирания строй (12-ТЕТ) са приблизително близки до някои интервали от точно темперирания строй. По специално – близки са точната кварта, квинта, терца и секста. Квинтите и квартите са незабележимо близки до точния строй. По-рядко приложим е равномерно-темперирания строй 19-ТЕТ, в който има по-добри терци и сексти, но слаби кварта и квинти, сравнен с 12-ТЕТ.

Име	Точна стойност в 12-ТЕТ	Десетична стойност в 12-ТЕТ	Центове	Точен интервал	Центове в Точно темпериран строй	Разлика
Унисон (C)	$2^{\frac{0}{12}} = 1$	1.000000	0	$= 1.000000$	0.0000	0
Малка секунда (C#)	$2^{\frac{1}{12}} = \sqrt[12]{2}$	1.059463	100	$\frac{16}{15} = 1.066667$	111.73	11.73
Голяма секунда (D)	$2^{\frac{2}{12}} = \sqrt[6]{2}$	1.122462	200	$\frac{9}{8} = 1.125000$	203.91	3.91
Малка терца (D#)	$2^{\frac{3}{12}} = \sqrt[4]{2}$	1.189207	300	$= 1.200000$	315.64	15.64
Голяма терца (E)	$2^{\frac{4}{12}} = \sqrt[3]{2}$	1.259921	400	$\frac{5}{4} = 1.250000$	386.31	-13.69
Чиста кварта (F)	$2^{\frac{5}{12}} = \sqrt[12]{32}$	1.334840	500	$\frac{4}{3} = 1.333333$	498.04	-1.96
Умалена квинта fifth (F#)		1.414214	600	$\frac{7}{5} = 1.400000$	582.51	-17.49
Чиста квинта (G)	$2^{\frac{7}{12}} = \sqrt[12]{128}$	1.498307	700	$\frac{3}{2} = 1.500000$	701.96	1.96
Малка секста (G#)	$2^{\frac{8}{12}} = \sqrt[3]{4}$	1.587401	800	$= 1.600000$	813.69	13.69
Голяма секста (A)	$2^{\frac{9}{12}} = \sqrt[4]{8}$	1.681793	900	$\frac{5}{3} = 1.666667$	884.36	-15.64
Малка септима (A#)	$2^{\frac{10}{12}} = \sqrt[6]{32}$	1.781797	1000	$\frac{7}{4} = 1.750000$	968.826	-31.91
Голяма септима (B)		1.887749	1100	$\frac{15}{8} = 1.875000$	1088.27	-11.73
Октава (C)	$2^{\frac{12}{12}} = 2$	2.000000	1200	$\frac{2}{1} = 2.000000$	1200.0	0

Таблица 4. Сравнение между точно темпериран и равномерно-темпериран строй [159]

ХАРМОНИЯ

Опитвайки се да дам определение за думата хармония⁸⁷, стигнах до извода, че красотата се ражда от хармонията. Това е така, защото хармонията е правилна пропорция – правилен ритъм. Животът е резултат именно от хармонията. Цялата Вселена е една симфония – всеки индивид в нея е нота, която се опитва да се настрои към ритъма и хармонията на Вселената. Музиката е миниатюра на хармонията в цялата Вселена. Както в останалите елементи на музиката, така и тук присъстват математическите закони.

Интересен факт засяга науката за индийската музика, която се оказва, че черпи вдъхновението си от три източника – математика, астрология и философия. Оказва се, че е така и при музиката на Запада, тъй като в нея цялата наука за хармонията и контрапункта се основава на математиката.

Друга интересен факт относно индийската музика откриваме в нейния превод на санскрит, според който науката за индийската музика се нарича *prestara*, което означава “математическо разположение на ритмите и ладовете”.

Хармонията е общо понятие, което изразява стройност, съразмерност, съгласуваност, баланс на елементи в общото.⁸⁸ Първоначално в музиката с хармонията се означават звуковисочинните отношения. Мелодия и хармония са тъждествени – т.е. добре направената мелодия изразява, притежава хармония. [16]

С появата на многогласието от 9 до 13 в. съдържанието на хармонията в музиката придобива нова същина – касае вертикалните аспекти, отнася се до многогласието. Целият етап от развитието на многогласието е наречен с общото название – полифония. Илюстрира отношенията между тоновете по вертикал, като в хармонията вече присъстват мерките консонанс и дисонанс – мерките за създадена хармония.

Следвайки в кратка хронология развитието на хармонията, стигаме и до следващият етап от развитието на хармонията, който се обуславя от раждането на жанра на операта през 1594 г., чрез операта “Дафне” от Джакомо Пери. Развива се инструменталната музика, а създадените в операта арии, каватини и речитативи с инструментален съпровод са причината за установяване на нов тип многогласие – хомофонията. Тя представлява главен глас със съпровождащ го глас – за разлика от

⁸⁷ Хармонията е наука за строежа, значението и взаимоотношенията на акордите /9.с.3/

⁸⁸ Според Боеций “Хармонията е съгласие на разногласието”

полифонията, където всички гласове са равнопоставени. Установяването на хомофонията е базата, върху която започва да се развива хармонията.

В технологичен смисъл благодарение на хомофонията се създават акорди, които изразяват хармонията. Оттогава насам възниква и представата за хармонията, като акордова система. Ако в древността хармония и мелодия са били тъждествени, то сега се когато някой каже „чуи тази хармония“, това се разбира, като „чуи тези акорди“.

Въпреки всичко, разликата между акордовата и хармонична структура е интересна не само от математическа гледна точка, но и от чисто слухово възприемателна психо-физиологична гледна точка. В тази връзка всеизвестна е дефиницията за хармонията, като „наука за строежа, значенията и връзките между акордите“, която в съвременната музика се измества от друга обобщаваща дефиниция, че хармонията е „звуквисочинна организация (координация) на музикалния материал, по хоризонтал и вертикал“.

АКОРД

Създадените акорди в епохата на хомофонията са познатите до днес, т.нар. терцови акорди.

Акордът като набор от интервали и числови съотношения рефлектира върху една от първите системи за обозначения на акордите – цифрования бас или т.нар. генерал бас. (Фиг.8)



Фигура 8. Тук, в баса тонът е С, а числата 4 и 6 индикират, че трябва да се изсвирят над този тон тоновете, които отстоят на кварта и секста от него – те са F и A

Първоначално както е известно, записът на акордите не е отразявал функционалните им тонални значения, нито степенната им принадлежност. Разбирането за акорда, като такъв и разбирането за ладо-хармоничните функции – дава

Жан Филип Рамо. На него дължим развитието на тоналната система на мажор и минор и функционалните значения на акордите.⁸⁹

Акордът се приема, като “централен, оперативен термин, свързан с многогласието – т.е. той съдържа и изразява многогласната тъкан”. Не всички съзвучия се определят като акорди, поради факта, че акордът се определя като такъв от определени структурни модели: акорди, образувани на единен интервалов принцип – терцов, квартов, секундов или акорди, образувани чрез интервалови модели. Тези акорди представляват вертикализация на определените хоризонтални модели.

В тази връзка, в структурата на много електронни музикални инструменти от голямо значение е акордовата функция. Функцията фиксира значенията на акордите, спрямо цялостната музикална материя. Тя е ролята, смисъла и значението на акордите по отношение на ладотоналността. Това е от значение понеже всеки фрактал е зависим от съответната ладотоналност.

При фракталите хармонизирането на дадена мелодия е едно решение на хармонична задача. Съзвучието и съгласуваността са в основата на тази задача, които алгоритъмът трябва да изпълни безпрекословно.

ФИБОНАЧИ И ЗЛАТНОТО СЕЧЕНИЕ

Ако разгледаме редицата от числа **1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...**, то можем да забележим, че всеки следващ член е равен на сумата от двата предходни. Тази редица наричаме *редица на Фибоначи*, а членовете ѝ - *числа на Фибоначи*. Ако означим n -тия член на редицата с F_n , то формалната дефиниция на редицата на Фибоначи е

$$F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2} \text{ за всяко } n \in N.$$

Ако разгледаме редицата от отношения на две последователни числа на Фибоначи

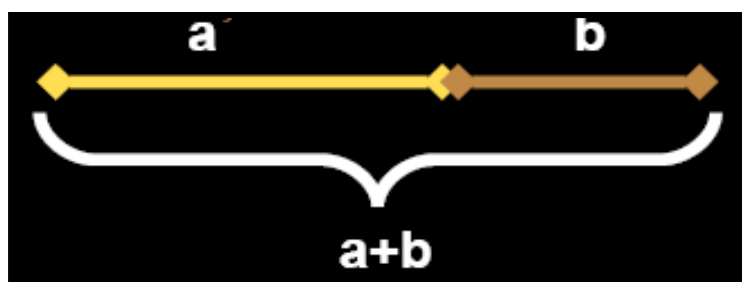
$$\frac{F_2}{F_1} = 1, \frac{F_3}{F_2} = 2, \frac{F_4}{F_3} = \frac{3}{2}, \dots, \frac{F_n}{F_{n-1}}, \dots,$$

то се оказва, че тя е сходяща и нека означим нейната граница с φ . (С други думи, всеки следващ член от горната редица се разполага по-близо до числото φ върху числовата ос.) След граничен преход в равенството $F_n = F_{n-1} + F_{n-2}$ достигахме до извода, че $\varphi^2 = \varphi + 1$, т.е.

⁸⁹ Функционалност на акордите – функционалността се обуславя от ладовостепенния състав на акорда. Според функционалността акордите се делят на устойчиви и неустойчиви /7.с.30/

$$\varphi = \lim_{n \rightarrow \infty} \frac{F_n}{F_{n-1}} = \frac{1 + \sqrt{5}}{2} \approx 1,61803.$$

Получената точна стойност на числото φ наричаме *Златно сечение*. Златното сечение (известно още като Златна пропорция, Златен коефициент или Божествена пропорция) е ирационално число, което може да се изрази и като отношение на части, за които по-малката част се отнася към по-голямата, така както по-голямата към цялото⁹⁰ (Фиг.9).



Фиг.9 Златната пропорция [160]

Оказва се, че числата на Фибоначи са част от естествената хармония, която е приятно да се усеща, приятно изглежда и приятно звучи.⁹¹ Неслучайно броят на тоновете и броят на белите и черни клавиши в една октава се изразяват с числа на Фибоначи, а именно - 5, 8, 13.

Малко музиканти осъзнават, че музиката основана на 8-степенна октава, както 1, 3, и 5-та степен, които създават основата на всички акорди, се базират на математическа зависимост. Благозвучните, хармонични акорди не са случайни. Най-важните хармонично звучащи интервали могат да се получат с помощта на отношенията на числата 1, 2, 3,5,8.

Йохан Кеплер открива подобна връзка и в интервалите между основните мажорни и минорни тонични тризвучия (1,3,5 или 1,5,8). Той забелязва, че обикновено кулминацията на мелодията е в точката на златното сечение и е с неговата обща продължителност приблизително 1,618.

Редица учени изследват творчеството на велики композитори, като Моцарт, Бетховен, Бах, Барток, Дебюси, Шуберт, Чайковски и откриват, че в по-голямата си

⁹⁰ C.C.Clawson, Mathematical Mysteries: The Beauty and Magic of Numbers (New York:Plenum Press, 1966), 126

⁹¹ J.T.Anderson and C.S. Ogilvy, Excursions in Number Theory (New York : Dover , 1988), 139-140

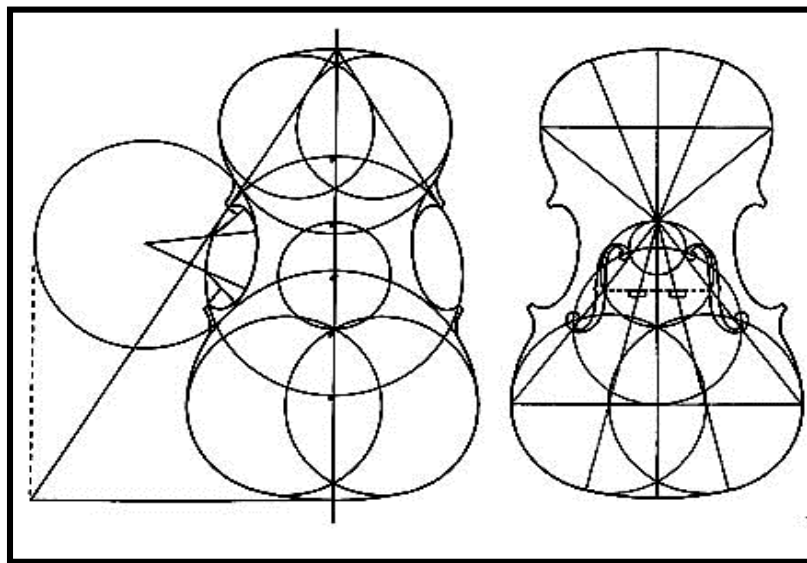
част, тези композитори интуитивно са използвали правилата на златното сечение, за да създават творбите си. Тази пропорция е очаровала много композитори през историята, които са я използвали основно като структурален апарат.⁹²

Един от най-ранните музикални примери за употребата на това е 40-гласовия мотет на Томас Талис *“Spem in alium”* : точно в момента на златната среда има един такт с абсолютна тишина, след който влизат едновременно всички 40 гласа. [127]

Бела Барток също широко използва златната среда в своята музика. Пример за това включва първата част на *“Музика за цигулки, перкусия и челеста”* – кулминацията се случва в златната среда, на 55 такт от общо 89 такта.

През 20 век Клод Дебюси използва тази пропорция в много от неговите творби, като например *“Reflets dans l'eau”*, където кулминацията се случва на 58 такт, от общо 94 такта ($58/94 \approx \phi$). [128]

Златното сечение се използва и при конструирането на музикалния инструмент цигулка (Фиг.11). То се използва при конструирането на f-фовете на цигулките на Страдивариус.⁹³

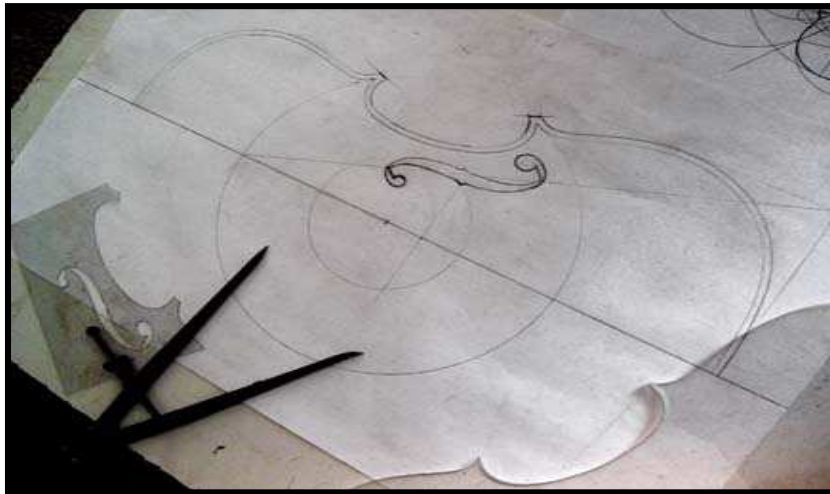


Фигура 11. Златното сечение върху цигулката [161]

Центрираните кръгове се изчертават според делението на златното сечение върху сегментовата линия, формирайки централната линия на тялото на инструмента.

⁹² Известни са думите на сестрата на Моцарт за гения "...Той винаги си играеше с числата и беше омаян от математиката..."

⁹³ В секция "Цигулката" на том Втори - "Нови Оксфордски Наръчник за Музиката", подробно е описан метода на конструиране, по подобие на Златното сечение



Фигура 12. Изчертаване на ефовете на цигулката [161]

Както може да се види (Фиг.12), това се използва, за да се установи мястото на четирите ъгла, както и мястото на f-дупките (ефовете). Този метод осигурява една цялостна схема за основните елементи от дизайна на един струнен инструмент като цигулката.

ПРИЛОЖЕНИЕ 2

ФРАКТАЛИТЕ КАТО МАТЕМАТИЧЕСКО ИЗРАЖЕНИЕ НА ПРОИЗВЕДЕНИЕ НА ИЗКУСТВОТО, СВЪРЗАНИ С МУЗИКАТА И ЖИВОПИСТА

За да докажа, че сходството между фракталите и музиката, фракталите и живописата, не се дължи само на повтаряемостта на музикалните форми, а е само още един подход за уеднаквяване на структурното им съдържание, ще обърна погледа си малко назад във времето. Времето останало в историята с наименованието класицизъм, с един от най-ярките представители на този период - композитора Йохан Себастиан Бах. Оказва се, че самият Бах е използвал интуитивно фрактални модели за композирането на някои свои произведения. Всеизвестен факт е че, музиката на Бах е трудна за моделиране и интерпретиране от други композитори. Като доказателство върху тези думи ще разгледам две от най-блестящите произведения на Бах – “Музикално приношение” и “Изкуството на фугата”. Това което илюстрира красотата и безкрайността на фракталното изкуство и същевременно ми даде насока да търся сходството между фракталите и музиката на Бах може да се види най-красноречиво на фигура 1, както и в Приложение 3, Fractali Slideshow.m4v.



Фигура 1. Фрактал [140]

МОДЕЛ НА ФРАКТАЛНОСТ В КЛАСИЧЕСКАТА МУЗИКА

Интересна асоциация, която може да се направи между фракталния модел и музиката, са две форми на музикална структура - канон и фуга. Оказва се че както в канона, така и във фугата, се откриват сходни черти в приципа на структуриране, които се откриват при създаването на фрактали. За целта ще разгледам някои от по-важните черти, свързани с канона и фугата.

КАНОН

Това което стои като идея зад структурата на канона е, че една тема се изпълнява на фона на самата себе си. Това се постига като отделните участващи гласове изпълняват копия на темата. Един пример за такъв канон е *“Три слепи мишки” (Three Blind Mice)*. В него темата се появява в първия глас и след известно закъснение се появява нейно копие в същата тоналност. След същото като времетраене закъснение спрямо втория глас се появява третия глас със същата тема и т.н. Повечето теми не могат да се хармонизират със себе си по този начин. За бъде една тема подходяща за канон трябва да може да изпълнява двойна, тройна или четворна роля – да бъде част от мелодията и да бъде част от хармонизацията на същата мелодия. Когато например в канона има три гласа, всеки тон от темата трябва да действа и като мелодия и в две различни хармонии. По този начин всеки един тон от канона има повече от едно значение.

Има и по-сложни канони, при които първото нарастване на сложността идва когато копията на темата са изместени не само във времето, но и във височина; тогава първият глас може да изпълнява темата, започвайки от до, а втория глас, застъпвайки се с първия, може да изпълнява същата тема, като започва от сол. Третият глас, който започва от ре, може да препокрие другите два гласа. Друго усложнение може да се получи ако темпата на двата гласа не са еднакви – вторият глас например се изпълнява два пъти по-бързо от или по-бавно от темпото на първия глас. Първият тип се нарича скъсяване, а втория удължаване.

Друг похват, който се използва в каноните е темата да се инвертира, което значи да бъде създадена мелодия, която прави преход надолу, докато оригиналната мелодия преминава нагоре, при това с точно същия брой полутонове. Бах например използва много този похват на инверсия.

Най-интересният вид копие е ракоходното копие, когато темата се изпълнява обратно във времето. Името идва естествено от движението на рака. Бах включва такъв вид ракоходно копие в «Музикално приношение».

Всеки вид копие запазва цялата информация от оригиналната тема, така че темата да може да бъде възстановена изцяло от което и да е копие. Подобна запазваща информация често се нарича изоморфизъм – сходен, еднакъв по форма.

Някои канони имат свободни гласове, които не следват темата на канона, а просто се хармонизират добре с гласовете, които се намират в канон помежду си. Всеки канон в «Музикално приношение» има за тема различен вариант на кралската тема, като много от горепосочените похвати са използвани в това произведение. Така например в един тригласен канон озаглавен *Canon per augmentationem contrario motu* (Канон с удължаване и обратно движение), средният глас е свободен – изпълнява кралската тема, докато другите два гласа се движат канонически над и под него, използвайки похватите на удължаване и инверсия.

ФУГА

Фугата⁹⁴ е музикален термин, който означава полифонична музикална форма, както и музикално произведение или част от него, написани в тази форма. Основата на фугата е една тема, която се разработва равностойно от всички участващи гласове. Обикновено се състои от три части:

1. В първата част, наричана експозиция, музикалната тема се представя с видоизменения от всички водещи гласове в основната за фугата тоналност, но също в доминантната и субдоминантната ѝ тоналност.

2. Във втората част, гласовете които в определен момент са в подчинено положение спрямо водещия глас, изпълняват мелодии, които са като музикална реплика на мелодията на водещата партия. В тази част на фугата много често се използва модуляция, преобразуване на темата, контрапункт.

3. В третата част, която понякога неправилно се нарича реприза, всъщност тази реприза е един от начините, по които може да се завърши фугата. Тази част е обобщение на музикалния материал от предишните части.

⁹⁴ от латински: fuga - бягство, надпревара

Съществуват сложни фуги, които се изграждат не върху една, а върху две и повече теми. За разлика от други циклични форми, фугата не предполага контраст на теми, а контраст в рамките на една тема. Фугата е подобна на канона, доколкото се основана на една тема, която се изпълнява от различни гласове и в различни тоналности, понякога в различно темпо. Но концепцията на фугата е много поразчупена от тази на канона. Отличителна черта на фугата е начинът, по който започва – с един единствен глас, който изпълнява темата. Когато той приключи, възпва втория глас, или с четири степени по-високо, или с три степени по-ниско. Междувременно първият глас продължава с изпълнение на контратема – това е втора тема, избрана да внесе ритмичен, мелодичен и хармоничен контраст на темата. Следващите гласове възпват последователно, всеки със своето изпълнение на темата – често под съпровода на контратемата в някой друг глас.

Двете фуги в «Музикално приношение» са изключителен образец в този жанр, които не биха могли да бъдат композирани по зададен шаблон. Те притежават нещо много по-дълбоко от стандартизираното звучене на фугата.

МУЗИКАЛНО ПРИНОШЕНИЕ

«Музикално приношение» се състои от една фуга в три гласа, една фуга в шест гласа, десет канона и трио-соната (Фиг.2). Фугата в шест гласа е едно от най-сложните произведения на Бах, а нейната главна тема е Кралската тема (Фиг.3).



Фигура 2. Откъс от Музикално приношение [162]

Кралската тема, дадена на фигура 3, е много сложна, ритмично неравномерна и силно хроматична. Трио-сонатата е много мелодична, почти танцувална, но също се основава на темата на краля.

Десетте канона в «Музикално приношение» са едни от най-съвършените канони, писани от Бах. Интересното е че Бах не ги завършва, понеже те са един вид загадки към крал Фридрих. По онова време е била обичайна т.нар. музикална игра, при която се дава една тема с няколко подсказки с различна степен на сложност, като някой друг е трябвало да открие канона, базиран на тази тема.



Фигура 3. Кралската тема [162]

КРАТЪК АНАЛИЗ НА КАНОНИТЕ ОТ “МУЗИКАЛНО ПРИНОШЕНИЕ”

CANON 1. Първият канон от «Музикално приношение» е “ракоходен” (*crab*): използва кралската тема с втори каноничен глас, в който темата звучи назад.

Този канон се сравнява с лентата на Мьобиус.(Фиг.14) Това е математическото изражение на музикалния език на Бах (Фиг.4). (*Приложение 3, J.S.Bach – Crab Canon on a Möbius strip.mp3; J.S. Bach - Crab Canon on a Möbius Strip.avi*)



Фигура 4. Първи канон [86]

CANON 2. Това е прост канон в унисон за две цигулки, като кралската тема се появява в независима линия под каноничните гласове (Фиг.5).



Фигура 5. Втори канон [86]

CANON 3. В третия канон каноничните гласове са инструктирани да се придвижват в противоположни ходове (*per Motum contrarium*), докато кралската тема се намира в най-високия глас (Фиг.6).



Фигура 6. Трети канон [86]

CANON 4. Бах описва този канон като *Notulis crescentibus crescat Fortuna Regis*, което се отнася до ритмично увеличение на следващия глас: "Както нарастват тоновете, нека силите на Краля правят същото." Обърнатият ключ сигнализира следващия го канон да се придвижи в обратната посока на водача (Фиг.7)



Фигура 7. Четвърти канон [86]

CANON 5 (Canon per Tonos). В канон 5 кралската тема се появява в най-високия глас, като каноничните гласове са разделени от интервала чиста квинта. В този канон Бах намерва за модулираща спирала, в която всяко повторение на канона нараства в тоналността с една цяла стъпка. (Както тоналностите нарастват, нека така расте и славата на Краля). Щефан Шелер (*Stefan Scheller*) определя този канон като пример за акустична илюзия, в която скалата може да бъде направена да изглежда така сякаш расте, докато това всъщност не се случва.



Фигура 8. Пети канон [86]

Първият глас в канона изпълнява вариант на кралската тема, а под него двата гласа внасят каноническа хармонизация на базата на друга тема (Фиг.8). Темата на пониския от двата гласа е в до минор, което е тоналността на канона, а по-високият глас изпълнява същата тема, изместена по височина с квинта. Интересното за този канон е че се отличава от всички останали по това, че когато завърши вече не е в основната си тоналност до минор, а е в ре минор. Бах успява неусетно да модулира тоналността. Канонът е изграден по такъв начин, че този своеобразен край да се свързва гладко отново с началото, като този процес може да се повтори многократно, модулирайки през много тоналности и пак да се върне в началото. Тези последователни модуляции отвеждат канона далеч от основната тоналност, но точно след шест такива модуляции оригиналната тоналност до минор се завръща. Всички гласове са точно една октава по-високо, отколкото са били в началото. Навярно намерението на Бах е било този процес на модуляция да продължи до безкрайност и затова вероятно написва «Както модуляцията се извисява, така нека се извисява и славата на Краля». Понеже канонът има потенциал за безкрайно повторение, може да се нарече «Безкрайно извисяващ се канон».

С този канон Бах ни дава пример за понятието «странен цикъл». Този цикъл се случва, когато се движим нагоре или надолу през нивата на някаква система и неочаквано се озовем там, откъдето сме тръгнали – в случая това е системата на музикалните тоналности. Понякога този цикъл може да бъде скрит, друг път да е явен, понякога може да е в правилно положение, друг път да е обърнат. Неслучайно този похват дава повод да се правят асоциации между безкрайния канон и литографията «Водопад» на Ешер (Фиг.20).

CANON 6. С десет такта между лидера и последователя, този канон е с канонична fuga с последовател на квинта. Най-високият глас е точно транспониране на средния отиващ нагоре в чиста квинта (Фиг.9).



Фигура 9. Шести канон [86]

CANON 7. Името което Бах дава на този канон е *perpetus*, което следва от факта, че този канон няма задоволителен край. Всъщност всеки канон, който е повтаряем, може да бъде считан за "*perpetual*." Интересна тук е употребата на басов ключ за последващия канон (Фиг.10).



Фигура 10. Седми канон [86]

CANON 8. Осмият канон технически не е един канон, а два канона, които се съединяват в такт 18. Каноничните гласове на втората половина са вертикалните огледални образи на техните противоположни части от първата половина (Фиг.11).



Фигура 11. Осми канон [86]

CANON 9. Лидерът в този канон прочита нотите на алтов ключ, докато следващия го канон чете нотите на басов ключ. Това също е огледален канон (Фиг.12). В този канон Бах не показва времеви интервал, приканващ ни да го открием сами - *Quaerendo invenietis* - "Търси и ще откриеш, искай и ще получиш."



Фигура 12. Девети канон [86]

CANON 10. Бах не специфицира времеви интервал, подобно на предишния канон, очаквайки от нас да потърсим и открием - *Quaerendo invenietis* (Фиг.13).

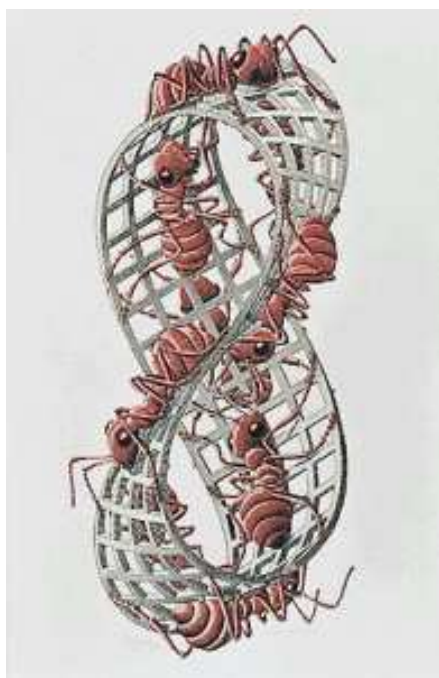


Фигура 13. Десети канон [86]

Съпоставяйки музикалния материал на „Музикално приношение“ с математическата същност на изразяването на фракталната музика виждаме много прилики в каноните от Музикално приношение на Бах.

В тази връзка можем да забележим в енигматичните **Канони 1 и 2** от **“Музикално приношение”**, как ръкописът показва единична партитура, чието начало се свързва с края. Това пространство е топологично еквивалентно на пакет от линейни сегменти в един кръг, познато като лентата на Мьобиус (Фиг.14). Едновременно изпълнение на дълбоко свързани линии, които се движат напред и назад, дава представа за два гласа, чиято симетрия се определя като обратима еволюция. Една музикална инверсия или обръщение се изгражда и след това се просвирва наобратно.⁹⁵

Една безкрайна кръгово-образна лента обикновено има две отделни повърхнини, една отвън и една отвътре. В тази лента девет червени мравки вървят една след друга и обхождат както предната, така и обратната страна. Въпреки това лентата има само една повърхнина.

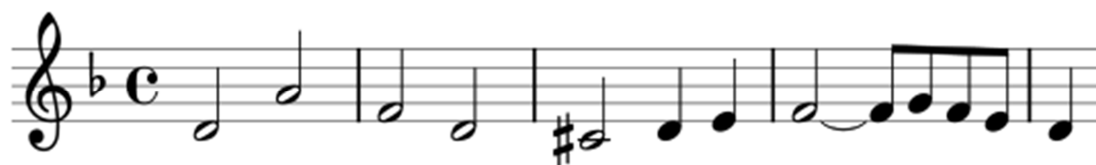


Фигура 14. Лента на Мьобиус [163]

⁹⁵ Приложение 3, J.S.Bach – Crab Canon on a Möbius strip.mp3; J.S. Bach - Crab Canon on a Möbius Strip.avi

ИЗКУСТВОТО НА ФУГАТА

Докато работи по «Музикално приношение» Бах решава да създаде творба, създадена изцяло по една проста тема – «*Изкуството на фугата*» (*Die Kunst der Fuge*), която Бах не успява да завърши. Тя се състои от 18 сложни фуги и канона (Фиг.15, 16). В този цикъл Бах използва всички инструменти и техники за писане на полифонични произведения. Фугите са озаглавени «Контрапункти». Последните четири фуги са свързани две по две така, че едната нота по нота е точна инверсия на другата, сякаш отразена в огледало.



Фигура 15. Всяка от 14-те фуги, с изключение на последната (незавършена) използват като тоналност d moll. [164]



Фигура 16. Контра-фуга, в която вариацията на основната тема се използва и в правилна и обърната форма [164]



Фигура 17. Огледална инверсия от Изкуството на фугата [164]

Огледалните фуги са двойки, в които каквото прави едната, нейното огледално отражение прави обратното (Фиг.17). Бах нарича оригиналната фуга *Rectus*, а обратната *Inversus*. В тези фуги Бах прилага не една, а шест огледални инверсии в едно и също време.

1. Мелодиите са огледални: всяка мелодия от огледалната фуга се придвижва в обратната посока. Където мелодията се изкачва, огледалната фуга слиза. Това се нарича мелодична инверсия или мелодично обръщение и се прилага към всеки глас от фугата.
2. Влизането на гласовете е огледално: в *Contrapunctus XII* всеки глас от огледалната фуга навлиза в обратен ред от оригиналната фуга. Където един глас навлиза пръв, в огледалната фуга влиза последен. Където един мотив се намира в партията на сопраното, огледалния мотив се намира в басовата линия. На фигурата може да се види влизането на гласовете в *Contrapunctus XII*.
3. Тоналните функции са огледални: тоническите тонове се заместват от доминанти, а доминантите от тоники. Това обръщане най-вече може да бъде доловено в педалните тонове. Където има педал на “до”, педала на огледалната фуга се намира на “сол” и обратно.
4. Тоналностите са огледални: модулациите преминават в обратната посока. Където има модулация в доминантата, огледалната фуга модулира в субдоминантата.
5. Последованията са огледални: последованията преминават в обратната посока. Където едно последование се изкачва нагоре, в огледалната фуга до отива надолу.

Огледални фуги са технически невероятни произведения. И в двете огледални фуги *Contrapunctus XII* и *Contrapunctus XIII*, първо се чува фугата, а след това нейната огледална фуга. Цялото произведение е обърнато наопаки. Гласовете и интервалите са прецизно спазени, но са завъртяни и обърнати. Партиите са завъртяни - басът от първата фуга става сопран в огледалната фуга, тенорът става алт, алтът става тенор, сопранът става бас. Цялата творба се състои от три фрази, всичките аранжирани сходно една на друга.

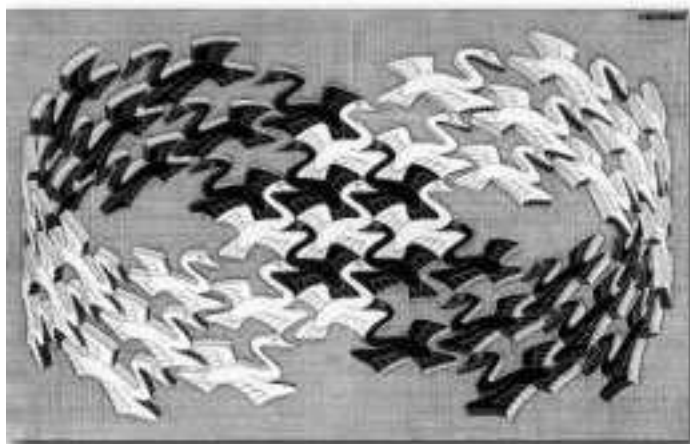
Както видяхме от разгледаните примери дотук, себеподобността и скалирането, които са отличителните черти на фракталите, са били използвани в много творби от минали векове. В този случай себеподобността се проявява чрез повтарящата се употреба на едни и същи мотиви в един голям откъс от произведението.

Себеподобността тук е частична, защото засяка мотивния материал на творбата, но не и неговата хармония или цялостната му форма.

ФРАКТАЛИТЕ, МУЗИКАТА И МАУРИЦ ЕШЕР

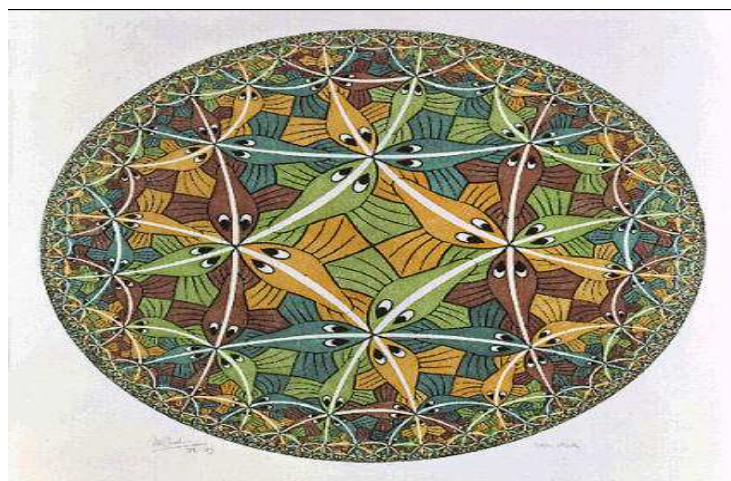
Всеизвестен е стремежа на твореца да търси аналогии между творчеството си и други форми на креативността. Водена от интереса си към връзката между музиката и математиката, математиката и живописата, вниманието ми бе привлечено от факта, че една от най-известните литографии на Ешер - *“Circle Limit III”* (Фиг.19), бива публикувана през 1959 г., почти 20 години преди Беноа Манделброт, откривателят на фракталите, да започне да ги изучава. Така започна и моето чисто философско търсене и анализиране на връзката между фракталите, музиката, математиката и живописата.

Това което несъмнено обединява фракталите, музиката, математиката и живописата е похвата рекурсия. Както е известно едно преброяване или изчисляване е абстрактното представяне на физичен процес и преливките между състояния или явления. Дефиницията за възможни състояния и събития се формулира като изчислителен модел. Една обратима система ще изчисли поставената функция два пъти: първият път ще изчисли движението напред, за да получи и съхрани изчислителния резултат, а втория път ще изчисли наобратно, като огледално-образно изчисление на обърнатата функция, преизчислявайки първата стъпка и връщайки системата към първоначалното ѝ състояние.



Фигура 18. "Swans", 1956 [163]

Рекурсията представлява една много обикновена концепция - нещо, което се повтаря и връща и има отношение към самото себе си. Отражението в едно огледало и после в още едно огледало е рекурсия, защото рефлектиращото огледало отразява своя собствен образ, правейки това безкрайно много пъти. Лебедите на фигура 18 показват този вид комбинация от математически модели и физика. Забележително е как лебедите са свързани много прецизно, със същото разстояние от близкостоящите лебеди и лебедите от следващия ред. Също така те се намират в много заворена линия, една конструкция която става възможна с рекурсията. Долупосочената творба на Ешер се нарича "*Circle Limit III*" (Фиг.19) и бива публикувана през 1959 г., почти 20 години преди Беноа Манделброт, откривателят на фракталите, да започне да ги изучава. На тази фигура може да се види, че образът има оригинал в средата, където всяка риба и перфектно съединена една с друга. Когато се вгледаме във фигурата виждаме, че образецът става по-сложен и по наситен в същото пространство, докато границата на кръга има безкрайни повторения от същия дизайн. Ако се увеличи една от тези миниатюри ще видим, че те продължават безкрайно. Това е есенцията на рекурсията и пример за фрактал.

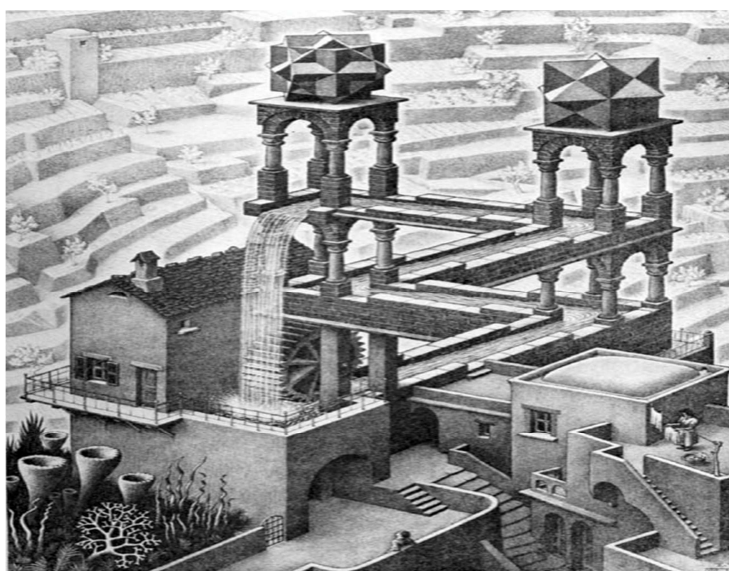


Фигура 19. Circle Limit III [163]

Много от образите на Ешер изразяват и показват същите характерни черти, които се срещат при фракталите. Поради тази причина намирам в творбите на Ешер интерпретиране на понятието странен цикъл, който се среща и в творбите на Бах. В творбите на Ешер се използват математически принципи на симетрия и закономерности. Например той използва топологически обекти като лента на Мьобиус, равнинни изображения на невъзможни в тримерното пространство обекти, изгражда

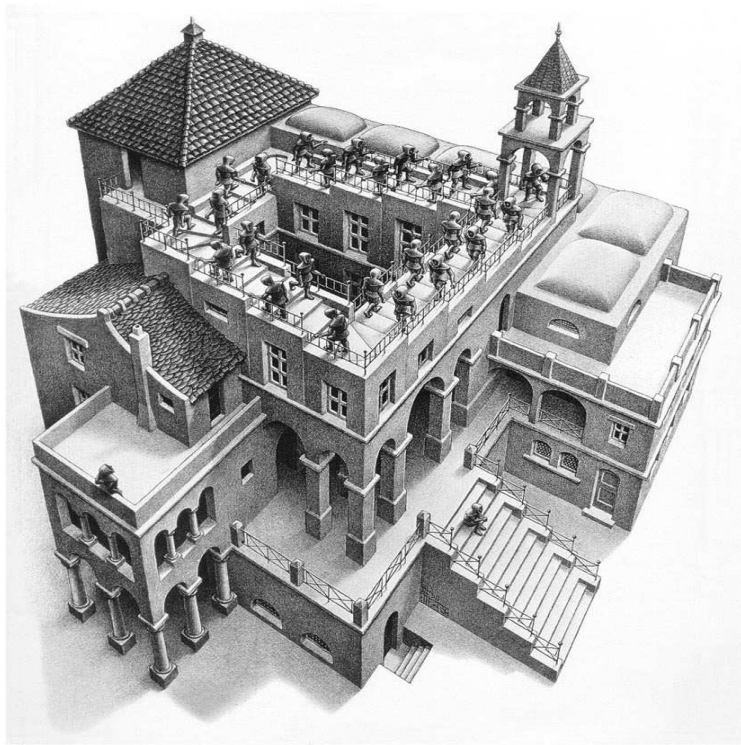
някои от картините си върху фрактали и триъгълника на Пенроуз, вплита крайни и безкрайни повърхнини, разгъвки на тела, които впоследствие съчетава с пейзажи, рисунки на животни и портрети. Изучавайки концепцията за представяне на безкрайността в двумерна равнина, Ешер илюстрира хиперболичните покрития в серията картини “Кръгов лимит” (*Circle Limit*, 1958-60, дърворез, Фиг.19).

През 1958 г. Ешер публикува статия “Правилното деление на равнината”, в която описва систематично как строй математическите обекти в творбите си.⁹⁶ Това ново отношение към изкуството Ешер претворява в картини като “Мьобиусов лист” (*“Moebius band”*, 1963, дърворез, Фиг.14), “Изкачване и слизане по сълби” (*“Ascending and Descending”*, 1960, литография, Фиг.21), “Водопад” (*“Waterfall”*, 1961, литография, Фиг.20).



Фигура 20. "Waterfal", 1961 [163]

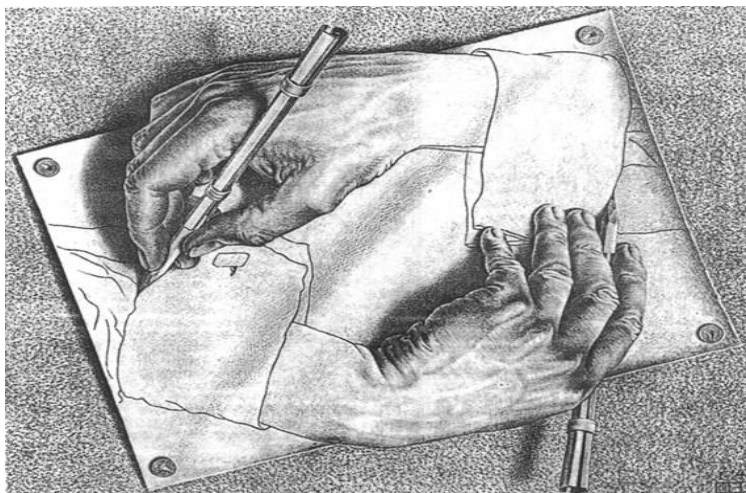
⁹⁶ Ешер подчертава, че “Математиците са отворили врата, която води към необятни територии”



Фигура 21. "Ascending and Descending", 1960 [163]

Типичната графика на Ешер съдържа много повече от проста симетрия или закономерност, защото много често има скрита идея. В частност, странныят цикъл е една от най-често повтарящите се теми в творчеството на Ешер. В литографията "Waterfal" (Фиг.20) и нейния безкрайно спускащ се цикъл от шест стъпки може да се сравни с безкрайно извисяващия се цикъл от шест стъпки в *Canon per Tonos*. Приликата е невероятна. Оказва се че Бах и Ешер изпълняват една и съща роля в две различни области на изкуството: музика и изобразително изкуство. Литографията "Ascending and Descending" (Фиг.21) представлява един вечен кръг, в който има толкова много стъпки, преди да се стигне отново до началната точка. Един по-стегнат цикъл се съдържа в "Waterfal" (Фиг.20), в който се съдържат само шест обособени стъпки.

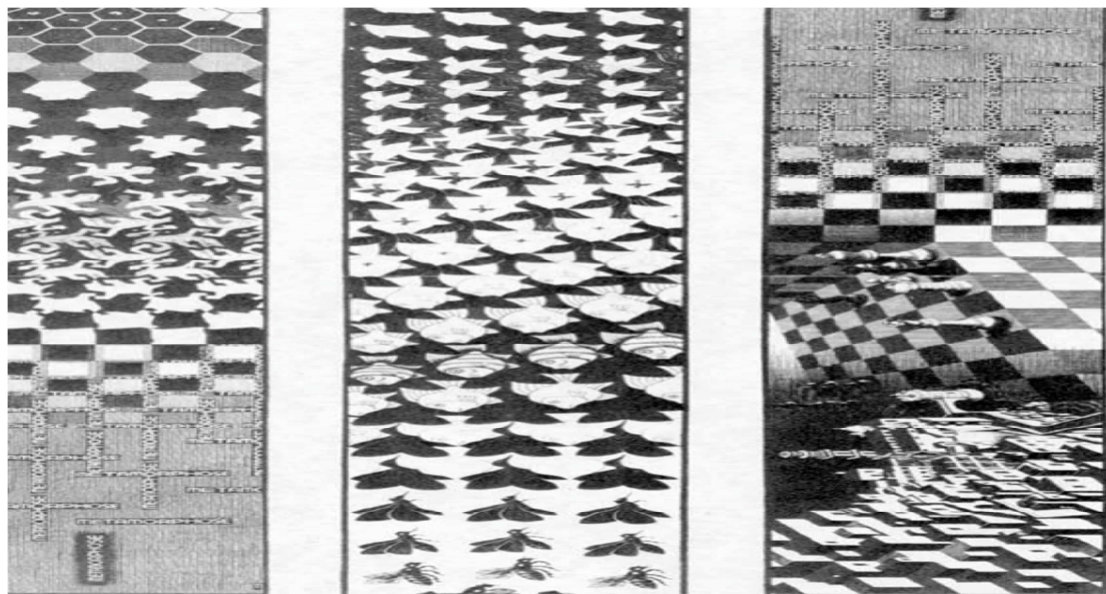
Действително броенето на нивата не само в картините на Ешер, но и във всички йерархични системи от много нива, се отличава с неизбежна неопределеност. В тази връзка не може да не се отбележи забележителната литография "Рисуващи ръце" ("Drawing Hands", Фиг.22), където всяка от двете ръце рисува другата, което представлява странен цикъл от две стъпки. Всичко в творчеството на Ешер притежава една основна идея, която е основна и за факталното изкуство - съдържа идеята за самото себе си.



Фигура 22. "Drawing Hands" [163]

В понятието за странен цикъл неявно се съдържа идеята за безкрайността, понеже цикълът е един вид способ да бъде представен един безкраен процес по краен начин. Именно безкрайността играе важна роля в много от рисунките на Ешер. Често копия от една тема се съчетават едно с друго и образуват визуални аналогии на каноните на Бах. Няколко такива мотива могат да се видят в гравюрата на Ешер "Метарморфоза" ("Metamorphosis II", Фиг.23). Тя прилича на Безкрайно извисяващия се канон, като се отдалечава все повече и повече от началната си точка и изведнъж се озовава отново там. В равномерно покритите с фигури равнини на "Метарморфоза" се забелязват чертите на безкрайността.

В разгледаните примери за сходство между творчеството на Бах и Ешер съществува една вечна недвусмисленост и безкрайност, които се коренят и в смисъла на странния цикъл. Най-вероятно математиците ще открият зад този факт присъствието на математиката, художниците ще отбележат присъствието на визуална интерпретация, а музикантите ще чуят музиката, която се крие зад толкова мистичност, цикличност и безкрайност. И както Бах и Ешер използват съвсем прости и интуитивно познати форми на движение, като нотната стълбица или движение по стълби нагоре и надолу, така и ние самите бихме могли да използваме въображението си, за да се разходим в техните безкрайно извисяващи се, циклични, припокриващи се, невъзможни светове.



Фигура 23. “Metamorphosis II”, 1940 [163]

РЕЧНИК

Алелес – (alleles) - един алел е една от две или повече форми на гена или група от гени.

Алеаторика (от латински: alea - зар) - модернистично течение в европейската музика, при което някои елементи от композирането са оставени на моментната преценка на изпълнителя, т.е. като цяло музиката е детерминирана, но в детайлите се подчинява на случайността. Заражда се в края на 50-те години на 20. век.

Атрактор – Атрактор (от англ. to attract – притеглям) – геометрична структура, характеризираща поведението във фазовото пространство в продължение на дълго време. Фазовото пространство е абстрактно пространство, координатите на което представляват степените на свобода на системата. Атракторът е област в пространството на възможните състояния, в която системата може да се движи, без да може да се откъсне от тях.

Вериги на Марков (Markov chain) – е математическа система, наименувана на Андрей Марков, която представя преходите от едно състояние в друго. Това е случаен процес, при който следващото състояние зависи само от текущото състояние, а не от изминалото. Веригите на Марков имат голямо приложение като статистически модели на реални процеси. Те се имплементират в алгоритмичната музика, конкретно в софтуерни програми като Csound, Max или SuperCollider.

Грапавина (ruggedness) – грапавина, неравност

Детерминирани и недетерминирани случайни процеси - недетерминирани случайни процеси са тези, при които всяка реализация представлява случайна функция на времето и неговите бъдещи стойности, които не могат да бъдат прогнозирани на базата на предишни измервания. Почти всички съществуващи случайни процеси в природата се отнасят към недетерминираните. **Детерминираните** случайни процеси са точно обратните процеси на недетерминираните.

Додекафония - музикална система, появила се непосредствено след атоналната музика, която се състои от 12 равноправни степени (хроматичната гама). За всяко музикално произведение се избира една 12 степенна редица (серия), която включва всички 12 тона от хроматичната гама, подредени произволно, с условието да не се повтаря тон, докато не се изреди цялата редица. Тази редица играе ролята на главна тема. Тя може да бъде пренесена от друг тон. Първата такава творба се появява през 1922 г., написана от Арнолд Шьонберг — Пет пиеси за оркестър. От тогава и до ден днешен тази техника се употребява от различни композитори.

Клетъчна автоматизация (Cellular automata) – дискретен модел, който се изследва в математиката, физиката и теоретичната биология. Състои се от симетрична решетка от

клетки, всяка от които има ограничен брой състояния, като "On" или "Off". Решетката има ограничен брой измерения.

Кросоувър (Crossover) - преминаване, пресичане, кръстосване. Има няколко значения спрямо материята, в която се използва:

- място, в което се пресичат няколко линии.
- кратък свързващ път, по който влак може да се прехвърли от една линия към друга.
- в генетиката – кръстосване на генетичен материал.
- в музиката – адаптация (преработка) на един музикален стил, смесвайки елементите на два или повече стилове. Действието кросинг оувър в един стил обикновено предоставя атрибути с няколко музикални стила и по тази причина често е привлекателен за по-широка аудитория.

Мапинг (mapping) или **Кодово-графично преобразуване** – операция, която свързва всеки един елемент от дадена поредица с един или повече елементи от втора поредица. Също така е графично представяне на една процедура, структура или система, която изобразява реда на връзките между различни компоненти и изчертава потоци от енергия, информация, материали. В музиката този термин се превежда като кодово-графично преобразуване, поради спецификата на процеса, който представя.

Риф (Riffology) – кратка, повтаряща се музикална фраза.

Стохастичен (Stochastic) (от гръцки *στόχος* за *цел* или *предполагам*) - термин, който се отнася до системи, чието поведение по същество не е детерминирано. Стохастичен процес е такъв, чието поведение не е детерминирано, като последващите състояние на системата се определят както от предвидими действия на процеса, така и от случайни елементи. **В музиката** стохастични елементи могат да бъдат генерирани от математически процеси. Стохастичната музика е предложена от Янис Ксенакис, който използва вероятностни функции, теория на игрите, теория на множествата. Преди появата на стохастичната музика, Джон Кейдж и други композитори създават та или неопределена музика, която се създава от случайни процеси, а не от математически модели.

Теория на хаоса (Chaos theory) е математически апарат, опериращ на базата на поведението на някои нелинейни динамични системи и описващ явление, известно като хаос или още известно с термина „чувствителност към началните условия“. Примери за подобни системи са атмосферата, турбулентните потоци, биологичните популации и икономическите системи. Въпреки че математическите системи с хаотично поведение се явяват детерминирани, т. е. се подчиняват на някакъв строг закон.

Фрактал (от латински означава счупен, надробен) – геометричен обект, който бива предложен през 1975 г. от Беноа Манделброт.

Фрактална геометрия – фракталната геометрия се описва изобразява чрез алгоритми – поредица от инструкции как да бъде създаден един фрактал.